VIRPIL CONTROLS
COMMUNITY SERVERS



VPC Link Tool

Руководство пользователя

VIRPIL CONTROLS

Содержание

УСТАНОВКА И ЗАПУСК 3
БАГИ И ПРОБЛЕМЫ 5
ПОЛЕЗНЫЕ ССЫЛКИ 5
Общие сведения 6
Главное окно 7
Окно редактирования правила 11
Правило Argument, окно Source 14
Правило Keyboard, окно Source 15
Правило Axis, окно Source 16
Правило Button, окно Source 17
Опции приложения 18
Общие настройки приложения 19
Настройки Elite Dangerous 21
Настройки War Thunder 22
Настройки Ил-2 23
Настройки DCS World и X-Plane 11 24
Генератор профилей LED-устройств 25
Начало работы 27
Профиль устройства без дочернего SLAVE 28
Профиль устройства с дочерним SLAVE 30
Создание LED правил 32
Правила аргументов 33
Интеграция с симуляторами 39
Интеграция с DCS World 39
X-Plane 11 Integration script 46
Il-2 Sturmovik simulations family integration 55
VIRPIL Devices LED Addresses 58

2 УСТАНОВКА И ЗАПУСК

УСТАНОВКА И ЗАПУСК

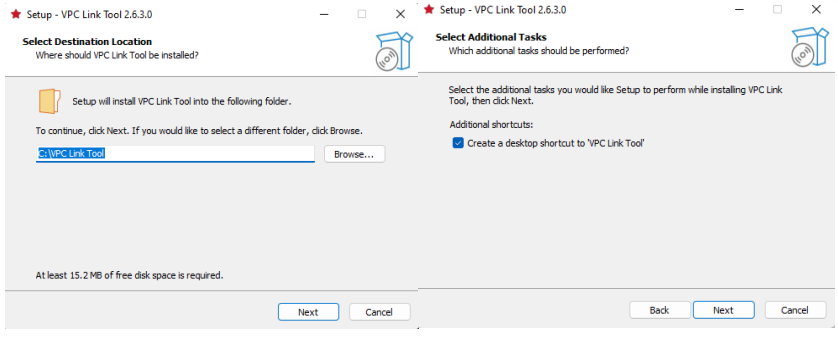
Для начала, откройте веб-страницу VIRPIL Updates Portal для загрузки и установки приложения VPC Link Tool.



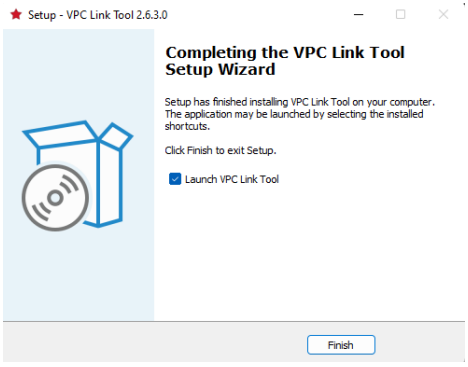
На данной веб-странице представлено 2 способа загрузки приложения:

- По нажатию кнопки MORE Вы будете переадресованы на страницу Link Tool, которая содержит несколько под-страниц (Description - Описание, Releases - Выпуски, Profiles - Профили). Выберите желаемую или крайнюю (latest) версию LT и нажмите кнопку скачать (download).
- По нажатию кнопки DOWNLOAD произойдет загрузка крайней версии приложения.

После загрузки установочного файла, запустите его. Вам будем предложен путь установки приложения и возможность создания ярлыка.



После окончания установки LT на экране появится интерфейс с возможностью немедленного открытия приложения (опция Launch Link Tool).



БАГИ И ПРОБЛЕМЫ

При возникновении каких либо проблем выполните полное удаление VPC Link Tool с Вашего компьютера, загрузите крайнюю стабильную версию приложения и создайте профили устройств заново. Процесс создания профилей LED устройств описан в следующих частях данного руководства.

Если Вы все ещё наблюдаете проблемы с приложением - обратитесь к специалистам в тех-поддержку VIRPIL Controls или посетите специальный форум VIRPIL Controls по адресу :

https://forum.virpil.com/index.php?/topic/2561-vpc-link-tool/

ПОЛЕЗНЫЕ ССЫЛКИ

VIRPIL Главная страница:
https://virpil.com/
VIRPIL Форум:
https://forum.virpil.com/
VIRPIL Портал обновлений:
https://updates.virpil.com/
VIRPIL Серверы:
https://virpil-servers.com/

Общие сведения

Link tool – это бесплатное приложение, разработанное для платформы Windows, которое служит для постановки различных цветов LED-диодов на базе устройств, разработанных VIRPIL Controls. Основная концепция LT – это интеграция со многими популярными авиационными симуляторами:

- Il-2 Sturmovik 1946 Забытые сражения
- Il-2 Sturmovik Скалы Дувра / Desert Wings
- Il-2 Sturmovik Великие сражения - *требует настроек конфигурации симулятора*
- War Thunder
- DCS World – *требует интеграционный LUA-скрипт*
- Microsoft Flight Simulator 2020 (MSFS2020) – *требует профиль переменных*
- Microsoft Flight Simulator X (FSX) - *требует профиль переменных*
- Microsoft Flight Simulator Deluxe Edition– *требует профиль переменных*
- Lokheed Martin Prepar3D v4/v5 – *требует профиль переменных*
- Elite Dangerous
- X-Plane 11 - *требует интеграционный LUA-скрипт*

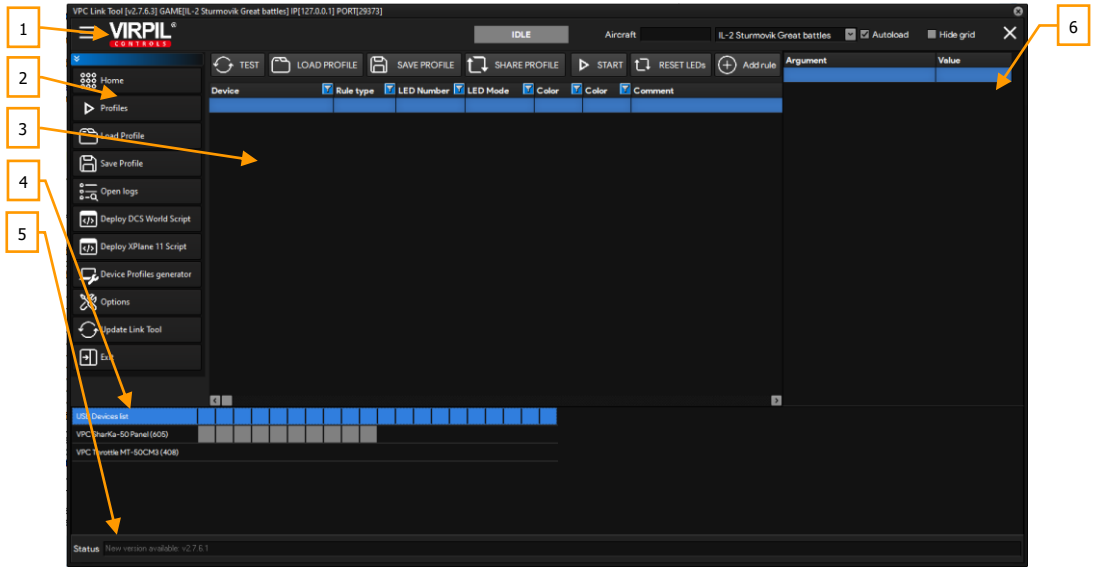
Приложение выполняет чтение данных из симулятора (различные значения состояния органов управления в кабине и телеметрия летательного аппарата) в виде ключ-значение и позволяет на основе этих данных создавать пользовательские правила управления LED диодами устройств VPC. Процесс интеграции приложения с различными симуляторами описан ниже, в данном руководстве пользователя, в разделе Интеграция с симуляторами.

Link Tool дополнительно дает возможность:

- создавать LED правила, основанные на нажатии различных комбинаций кнопок на клавиатуре;
- создавать LED правила, основанные на положениях различных осей любого USB-устройства;
- создавать LED правила, основываясь на состоянии кнопок, доступных для нажатия на любом USB-устройстве (кнопка нажата или не нажата).

Главное окно

Главное окно приложения состоит из шести частей:



1. Верхняя панель. Данная часть интерфейса приложения содержит кнопку управления меню (три горизонтальные линии, расположены в левой части панели), логотип компании, окно статуса приложения (серый индикатор статуса IDLE – простой приложения, см. картинку выше), поле с названием летательного аппарата (не редактируемое поле, заполняется исходя из данных некоторых симуляторов), выпадающее меню выбора симулятора, 2 быстрые опции (Autoload – автозагрузка, Hide Grid – спрятать таблицу аргументов) и кнопка X (закрыть приложение).
    - Опция Autoload используется для включения/отключения автозагрузки профилей LED-правил. Если опция включена, приложение считывает поле названия летательного аппарата и выполняет попытку загрузки профиля правил из файла *[ЛА].led.json* , который должен быть расположен по пути *LT/autoload*. Если такой файл отсутствует, процесс автозагрузки прерывается.
    - Опция Hide Grid используется для того, чтобы спрятать таблицу аргументов, которая расположена в правой части главного окна приложения, для улучшения общей производительности и стабильности во время выполнения расчетов LED правил. После старта приложения механизм расчетов LED правил периодически запускает множество функций, скрытых «под капотом» приложения. Поскольку процесс сам по себе периодический – приложение требует достаточное количество ресурсов для обновления интерфейса, в данном случае, таблица аргументов. Это довольно трудоемкая процедура. В случае, если приложение «фризит» или зависает, данная опция может помочь убрать или уменьшить проявление данных эффектов. При

включенной опции – таблица с аргументами будет скрыта, а обновление значений аргументов в ней будет отключено, приложение будет считывать значения аргументов из памяти.

Существует два симулятора, интеграция которых с LT требует дополнительных файлов, написанных на языке LUA-script – это XPlane11 и DCS World. Базовые шаблоны скриптов могут быть загружены с приложения напрямую, используя кнопки Deploy в левом боковом меню.

Все симуляторы семейства MSFS для интеграции с приложением требуют создания дополнительного файла с профилем переменных:

- Microsoft Flight Simulator X
- Microsoft Flight Simulator Deluxe Edition
- Microsoft Flight Simulator 2020
- Lokheed Martin Prepar3D v4/v5.

Для интеграции приложения с симулятором Elite Dangerous требуется указать путь к статус-файлу (Status.json), используя страницу опций.

Интеграция семейства симуляторов Il-2 Sturmovik требует дополнительных настроек в конфигурационных файлах, в зависимости от версии Il-2.

Создание профилей правил, устройств, деплой интеграционных скриптов и прочее мы рассмотрим в последующих разделах данного руководства.

2. Боковое меню. Данный блок интерфейса – это классическое меню приложения (может быть спрятано). Меню содержит следующие ссылки-функции:
   I. LOAD PROFILE – данная кнопки используется для открытия диалогового окна загрузки ранее сохраненных профилей пользовательских LED правил.
   II. SAVE PROFILE – данная кнопка используется для сохранения текущего набора LED правил (профиль) в файл с расширением **.led.json*. Если поле с названием ЛА не пустое, новый профиль LED-правил будет назван, используя шаблон: *[ЛА].led.json*. При желании название файла профиля может быть отредактировано в соответствующем поле диалогового окна.
   III. OPEN LOGS – данная кнопка используется для открытия файла лог-сообщений. Логи сохраняются автоматически, если приложение находится в режиме отладки (DEBUG).
   IV. DEPLOY DCS WORLD Script – данная кнопка используется для загрузки (с официального репозитория VIRPIL) и установки последней версии интеграционного скрипта с симулятором DCS World (скрипты используются для экспорта телеметрии летательных аппаратов из DCS в приложение).
   V. DEPLOY XPLANE 11 Script - данная кнопка используется для загрузки (с официального репозитория VIRPIL) и установки последней версии интеграционного скрипта с симулятором XPlane 11 (скрипты используются для экспорта телеметрии летательных аппаратов из XPlane в приложение).
   VI. DEVICE PROFILES GENERATOR – данная кнопка выполняет открытие диалогового окна генератора профилей LED устройств. Для каждого

устройства, которое оборудовано LED диодами и определяется приложением LT, должны быть созданы специальные профили, описывающие количество LED диодов, адреса диодов и номера команд, в зависимости от типа устройства. Процедура создания профилей устройств описана ниже в данном руководстве.

VII. OPTIONS – данная кнопка используется для вызова страницы опций, которая содержит общие настройки приложения, например, режим отладки Debug, автозагрузка профилей Autoload, данные соединения с различными симуляторами, (хост, порт), цвет диодов по умолчанию и прочие.

VIII. UPDATE LINK TOOL – данная кнопка используется для принудительной проверки наличия новой версии LT. Каждый раз при запуске приложение автоматически выполняет проверку наличия новой версии. Но если Вы желаете сделать это вручную в любой момент времени – данная кнопка поможет Вам это сделать.

IX. EXIT – данная кнопка используется для закрытия приложения. При закрытии приложения происходит восстановление цветов всех диодов в исходное состояние (согласно данным прошивки).

3. Список LED-правил. Это таблица с возможностями фильтрации, которая содержит список правил установки цветов на LED-диодах устройств, основываясь на позиции оси (AXIS), состояния какой-либо кнопки на USB устройстве, состоянию нажатия определенной комбинации кнопок клавиатуры (KEYBOARD) или состоянию аргумента симулятора.

Верхняя строка, находящаяся над таблицей, содержит несколько кнопок:

I. TEST – данная кнопка одноразово запускает итерацию расчета цвета диода, которая состоит из 4х этапов:
   - Запрос данных. На этой стадии приложение выполняет запрос данных от симулятора (если требуется).
   - Получение данных. На этой стадии приложение преобразует полученные данные в зависимости от протокола обмена между симулятором и отображает эти данные в таблице аргументов.
   - Расчет цветов LED-диодов. На этом этапе приложение выполняет построчное чтение правил из таблицы, в зависимости от статуса правила (валидное, не валидное правило) происходит генерация нового цвета диода (если требуется). Если правило определено как Валидное, соответствующая ячейка в таблице LED-диодов будет подсвечена определенным цветом (задается с помощью правила). Ячейка представляет собой адрес диода, расположенного на устройстве.
   - Установка цветов. На данном этапе приложение выполняет проход по списку LED устройств, посылая запросы к устройству на включение/изменение цвета диодов.

II. LOAD PROFILE – та же функция, что и в левом боковом меню.

III. SAVE PROFILE – та же функция, что и в левом боковом меню. Быстрая комбинация клавиш - LCTRL + S.

IV. START. Данная кнопка запускает приложение в активный режим.

V. STOP. Данная кнопка переводит приложение в режим простоя (остановка).

VI. RESET LED's. Данная кнопка выполнят сброс всех цветов диодов к цвету, согласно настройке приложения.

VII. ADD RULE. Добавляет новое пустое правило в таблицу.

Таблица, находящаяся в данной секции, представляет собой список пользовательских правил постановки цветов на диоды различных VPC устройств. Таблица состоит из следующих колонок:

- Device. Данная колонка содержит название устройства.
- Rule Type. Данная колонка отображает тип правила. Существует несколько типов правил: Argument (Аргумент), Button (Кнопка), Axis (Ось), Keyboard (Клавиатура).
- LED number. Данная колонка отображает номер диода (адрес на плате), к которому применяется данное правило.
- LED mode. Данная колонка отображает заданный режим работы диода. Существует 2 типа работы диодов в приложении: Steady (режим постоянной подсветки) или Flashing (режим «моргания»).
- Color #1. Данная колонка является визуальным представлением цвета, которым будет светить диод. Данное значение цвета используется в режиме работы диода STEADY.
- Color #2. Данная колонка – это визуальное представление второго цвета диода. В режиме Flashing сначала применяется Цвет 1, затем Цвет 2. Смена цветов выполняется последовательно. Если в правиле задан режим диода STEADY – данный параметр Color #2 не используется.
- Comment. Данная колонка используется для комментариев. Здесь можно оставлять различную информацию, например короткое описание работы правила.
- Status. Данная колонка отображает текущий статус правила (Enabled - Включено/Disabled - Отключено). Если статус задан как Disabled с помощью специального флажка на странице редактирования правила, приложение будет игнорировать такое правило на каждой итерации цикла работы.

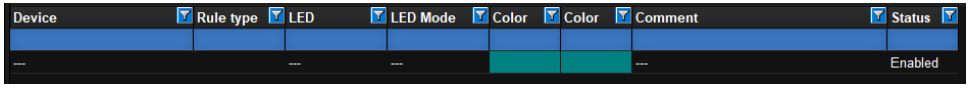
4. Список устройств и диодов. В этой секции отображается список VPC устройств, обнаруженных приложением. Таблица состоит из специальных ячеек, которые нумеруются слева на право, содержащие визуальное представление состояния диодов в реальном времени (если приложение запущено в режим работы). На картинке выше приложение определило, что к ПК подключена панель VPC SharKA-50 (Control Panel 3). Справа от названия устройства отображены 10 серых ячеек, поскольку приложение определило профиль LED-устройства (создается пользователем вручную), который содержит информацию о 10 LED-диодах и их физических адресах на плате (Серый цвет задан в качестве цвета по умолчанию в настройках OPTIONS приложения):
    - Прозрачный цвет ячейки: LED - диод не определен приложением.
    - Цветная ячейка: LED-диод определен приложением (профиль устройства был создан ранее). Цвет по умолчанию, который можно изменить в настройках приложения, применяется к каждой ячейке по умолчанию как исходное состояние диода.
5. Строка состояния. Данная строка отображает различную информацию, предупреждения и ошибки приложения в реальном времени.
6. Таблица аргументов. Данная таблица содержит в себе данные телеметрии в виде пары ключ/значение, переданные симулятором в приложение. Левая колонка таблицы содержит название аргумента (некоторые симуляторы позволяют изменять данное

значение), правая колонка содержит значение соответствующего аргумента (в случае, если данные корректно переданы из симулятора).

Окно редактирования правила

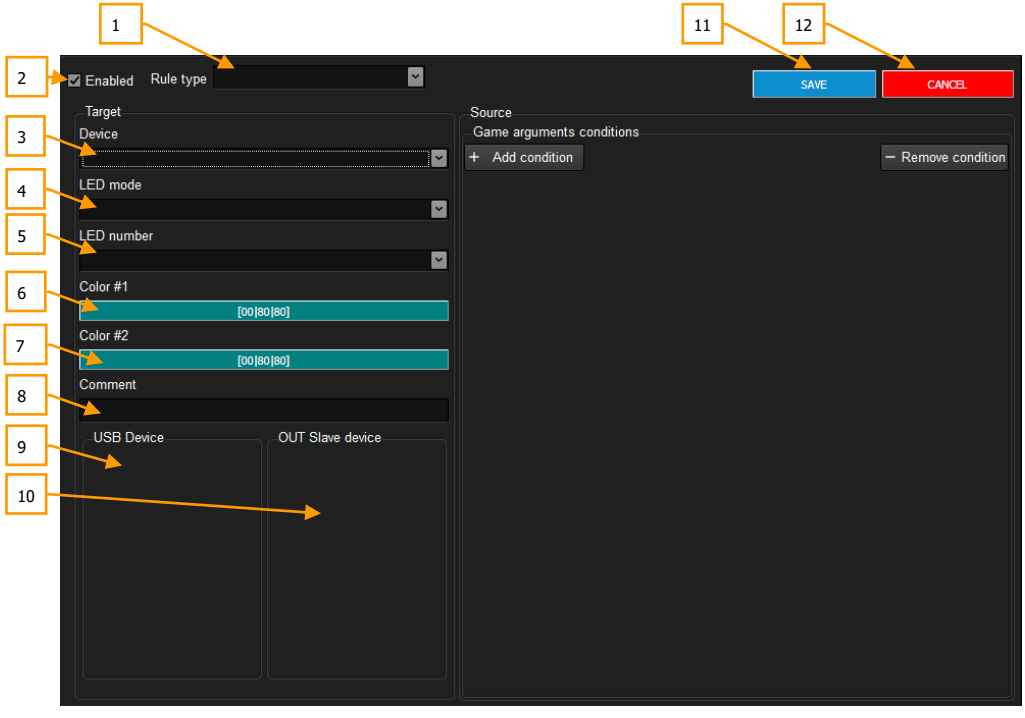
Данная часть интерфейса приложения позволяет осуществлять редактирование одного правила.

Нажмите кнопку ADD RULE чтобы добавить в таблицу первое пустое правило.



В таблице появится новая строка, некоторые значения ячеек в которой будут содержать тройной дефис «---». Такое правило является пустым и требует редактирования. Выполните двойной нажатие ЛКМ на новой строке для перехода к редактированию правила.

На картинке ниже представлен интерфейс редактирования правила.



Данная секция разделена на 2 части: Target (Целевое устройство) и Source (Источник).

Секция Target состоит из нескольких полей ввода данных, которые используются для определения целевого устройства и его диода, состояние которого будет изменено (если правило валидное).

Содержимое секции Source зависит от выбранного типа правила (rule type) и используется для определения условий изменения состояния заданного диода на целевом устройстве в секции Target. Эта секция используется для определения комбинаций кнопок клавиатуры, условий и значений различных аргументов ЛА, состояния физических кнопок на различных USB-устройствах, а также, условий и значений состояния осей различных USB-устройств.

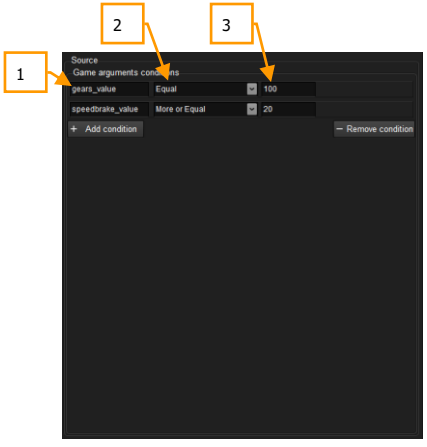
1. Выпадающий список Rule Type. Данный список используется для определения типа правила. Существует несколько типов правил, поддерживаемых приложением:
    - ARGUMENT. Данный тип правил используется для создания правил подсветки диодов, основываясь на значениях аргументов и их значений, переданных из выбранного симулятора в приложение. В окне Source на картинке выше вы можете видеть отображение списка условий аргументов, который выводится по умолчанию. Кнопка Add condition используется для добавления строки условия. Кнопка Remove condition используется для удаления первой и последующих строк условий сверху вниз. Механизм работы условий определяется с помощью следующего паттерна: **[УСЛОВИЕ] И [УСЛОВИЕ] И [УСЛОВИЕ]**. Например: *Условие1 И Условие2 И Условие3 И … УсловиеN*. В данном случае для того, чтобы диод изменил свое состояние, как задано в секции TARGET – все три условия должны быть валидны одновременно. В данном окне можно создать максимум 13 условий одновременно.
    - KEYBOARD. Данный тип правил используется для подсветки диодов в зависимости от нажатия различных комбинаций кнопок на клавиатуре. Например: если нажата комбинация LCtrl + H – диод #3 может «мигать» ярким зеленым цветом.
    - AXIS. Данный тип правил используется для изменения цвета диода в зависимости от положения оси на выбранном USB-устройстве. Для такого типа правил существует 3 типа возможных условий значений оси: значение оси больше или меньше (is more/ is less, в %, от 1 до 100) и в диапазоне ОТ и ДО (в %).
    - BUTTON. Данный тип правил используется для изменения цвета диода выбранного устройства в зависимости от состояния определенной кнопки, выбранной на любом из доступных USB-устройств в приложении. Например: если Кнопка3 на геймпаде нажата – установить постоянный Красный цвет диода на устройстве VPC SharKA-50 Control Panel 3.
2. Флажок Rule status. Данный флажок используется для включения/отключения правила.
3. Выпадающий список Device. Это список устройств VIRPIL, определенных приложением, цвет диодов на которых можно изменять.

4. Список LED Mode. Данный список используется для выбора режима диода. Существует 2 типа режимов диода – Steady (Постоянное свечение диода) и Flashing (Переменное свечение диода). В режиме Steady диод будет постоянно светить цветом, заданным в параметре Color #1. В режиме Flashing диод будет попеременно менять цвет диода со значения Color #1 на значение Color #2 в каждой итерации приложения.
5. Список LED number. Этот список используется для выбора диода из доступных, согласно ранее созданному профилю устройства. Если список диодов пустой – возможно профиль выбранного устройства не создан или не выбрано желаемое устройство, см. пункт 3.
6. Опция Color #1. Данная опция используется для выбора Цвета №1 в качестве основного, который будет применен к диоду в режиме Steady, и для определения первичного цвета в режиме Flashing («мерцание»).
7. Опция Color #2. Данная опция используется для определения вторичного цвета режима Flashing.
8. Поле Comment. Данное поле используется для ввода пользовательской информации относительно правила (например: описание аргумента или заданная комбинация на клавиатуре).
9. Изображение USB Device. В данном окне выводится изображение выбранного устройства в верхней части секции Target.
10. Изображение OUT Slave. В данном окне выводится изображение дочернего устройства, определенного прошивкой и профилем LED как SLAVE, выбранного как целевое устройство в секции Target.
11. Кнопка SAVE используется для сохранения изменений правила. Изменения записываются в таблицу. После сохранения изменений правила в таблицу – происходит автоматическое закрытие интерфейса редактирования правил и переход на главный интерфейс приложения.
12. Кнопка CANCEL прерывает редактирование правила и выполняет переход на главный интерфейс приложения без сохранения изменений правила в таблицу.

Правило Argument, окно Source

Данная секция используется для определения списка условий значений аргументов, полученных из выбранного симулятора. До 13 различных условий (на данный момент) может быть создано пользователем.

Каждое условие состоит из 3х редактируемых полей:



1. Название аргумента (строковое значение). Данное поле - редактируемое.
2. Условие (список предопределенных строковых значений). Данное поле – не редактируется.
3. Значение аргумента (числовое поле). Данное поле может быть отредактировано и должно содержать целочисленные значения или значения с плавающей точкой (1 или 1.0).

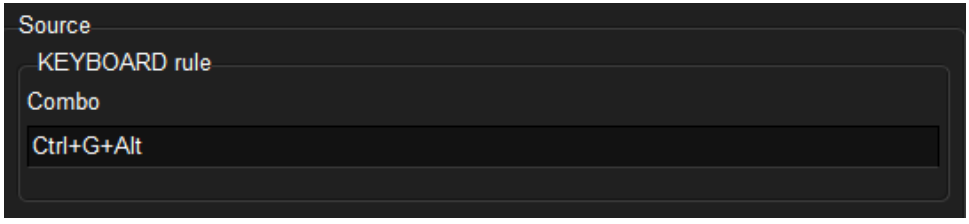
На изображении выше представлено 2 условия для аргументов gears_value и speedbrake_value. В данном случае правило будет валидно лишь тогда, если значение аргумента gears_value будет равно 100 И значение аргумента speedbrake_value будет БОЛЬШЕ ИЛИ РАВНО (MORE or EQUAL) 20. Оба условия должны работать одновременно.

В целом, для каждого условия пользователю доступно 6 логических операторов на выбор:

- Less - Меньше
- More - Больше
- Equal - Равно
- Not Equal – Не равно
- More or Equal – Больше или равно
- Less or Equal – Меньше или равно

Правило Keyboard, окно Source

Данная часть интерфейса используется для ввода комбинации кнопок клавиатуры.

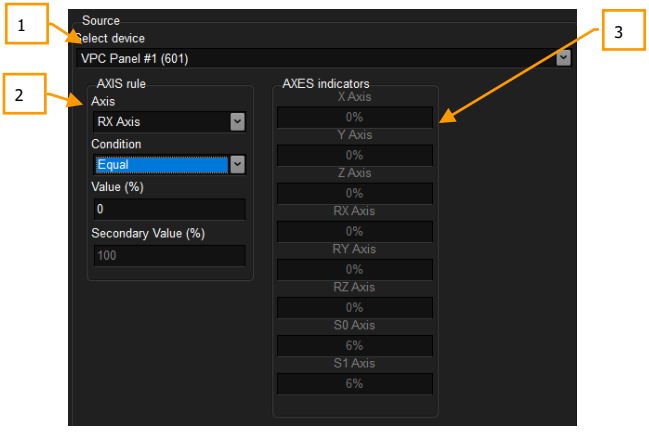


Нажмите ПКМ чтобы установить курсор (фокус) в поле ввода Combo, чтобы задать новую комбинацию. Если поле Combo вне фокуса – нажатая комбинация кнопок не будет определена и корректно сохранена.

Замечание! Приложение не эмулирует нажатия кнопок на клавиатуре. Правило будет валидным в том случае, если заданная комбинация была с помощью клавиатуры (не на джойстике, геймпаде и прочее).

Правило Axis, окно Source

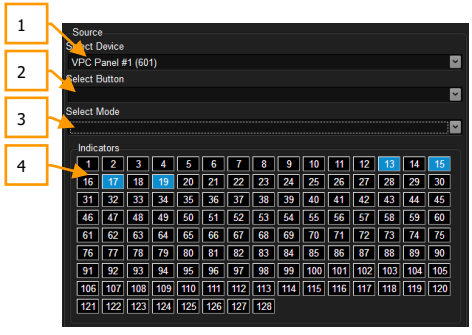
Данная часть интерфейса редактирования правила используется для выбора Оси и условия изменения цветов диода относительно выбранной оси и её текущих значений. Окно состоит из:



1. Выпадающего списка выбора устройства. Это меню содержит список всех доступных USB-устройств, определенных системой. Чтобы выбрать ось – нужно сначала указать устройство, на котором расположена желаемая ось.
2. Блок условия Оси. Данная часть интерфейса используется для выбора конкретной оси, условия постановки заданного цвета диода, а также, первичное и вторичное значение оси (вторичное значение используется при условии BETWEEN – В диапазоне ОТ и ДО). Меню Condition содержит аналогичный список типов условий, что и правило типа Argument. В качестве дополнения существует ещё один тип условий - BETWEEN. В случае выбора данного типа – при выполнении проверки условия будут учитываться первичное (Value) и вторичное (Secondary) значения оси. Все другие типы условий не учитывают вторичное значение Second Value.
3. Визуальные Осевые индикаторы. Данная часть интерфейса служит для отображения текущего положения оси в процентном соотношении. При изменении позиции осей на джойстике значения обновляются в реальном времени.

Правило Button, окно Source

Данная секция используется для создания правила установки цвета диодов, основываясь на состоянии выбранной кнопки на любом из доступных системе USB-устройств. Существует два типа состояния кнопки: Pressed (нажата), Released (не нажата).

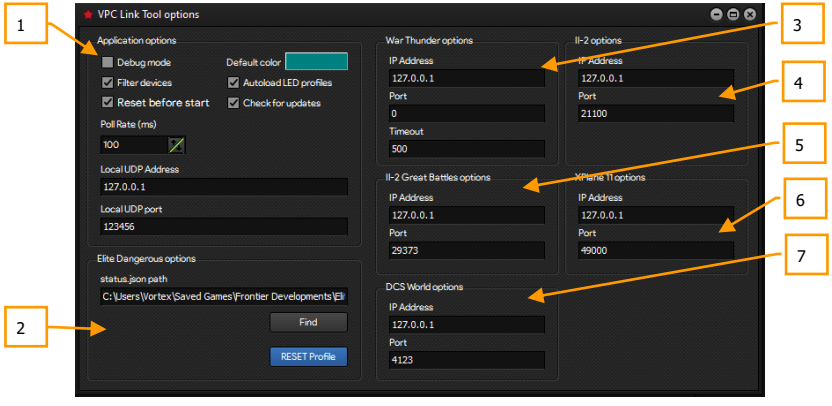


1. Окно выбора устройства, с которого нужно считывать состояние кнопки. Это список всех USB-устройств, обнаруженных приложением.
2. Меню выбора кнопки на устройстве. Данный список кнопок зависит от выбранного устройства. Если список пуст – возможно устройство не выбрано или приложение не может определить доступные кнопки на устройстве.
3. Выбор типа состояния кнопки.
4. Визуальные индикаторы состояния кнопок.

Примечание! Приложение не поддерживает считывание состояния комбинаций, только одной кнопки в то же время.

Опции приложения

Данный интерфейс приложения представлен как отдельное окно. При вызове окна опций происходит остановка функций приложения, при этом, само приложение остается открытым, но доступ к основным функциям основного интерфейса - заблокирован. Окно опций открывается с помощью комбинации *LCtrl + O*.



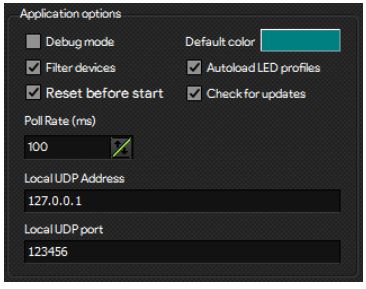
Интерфейс опций приложения состоит из:

1. Общих настроек приложения
2. Секции настроек интеграции Elite Dangerous
3. Секции настроек интеграции War Thunder
4. Секции настроек интеграции со старыми версиями Ил-2
5. Секции настроек интеграции с Ил-2 Великие Сражения
6. Секции настроек интеграции с X-Plane 11
7. Секции настроек интеграции с DCS World

Важно! После закрытия данного окна – сохранение всех внесенных изменений в настройки приложения и прочие – происходит автоматически.

Общие настройки приложения

Данный раздел опций используется для конфигурирования приложения в целом. Рассмотрим все доступные опции более подробно:



- Опция Debug mode. Данный флажок используется для включения режима отладки, после чего потребуется перезагрузка приложения. После перезагрузки появится окно терминала Windows CLI (Command-Line Interface). Это обычный терминал ОС, только в режиме чтения. Приложение будет осуществлять вывод системной информации в данное окно. Это довольно полезный инструмент при обсуждении проблем с технической поддержкой. Примечание! Режим отладки существенно влияет на общую производительность приложения. **Не рекомендуется использовать режим отладки для повседневной работы с приложением**.
- Опция Filter devices (не используется). Данная опция использовалась для включения фильтра USB-устройств и вывода лишь устройств производства VIRPIL. Данная опция устарела и будет удалена в будущем.
- Опция Reset before start. Данный флажок используется для включения-отключения функции сброса цвета всех диодов перед стартом работы приложения на цвет, заданный с помощью опции Default color.
- Опция Poll Rate. Данная опция отвечает за скорость опроса симуляторов, расчета правил и непосредственно, постановки цветов диодов. Чем меньше время опроса – выше скорость расчета данных и цветов диодов, и наоборот.
- Опция Default color. Данная опция используется для выбора цвета диодов по умолчанию перед началом старта синхронизации с симулятором. Цвет диодов учитывается лишь при активированном флажке RESET BEFORE START.
- Опция Autoload LED profiles. Данная опция используется для автоматический загрузки профилей пользовательских файлов-правил, основываясь на названия летательного аппарата. К сожалению, не все симуляторы поддерживают вывод имени выбранного ЛА в таблицу аргументов. Автозагрузка профиля происходит в момент, когда приложение обнаруживает изменения значения поля Aircraft, которое находится в верхней горизонтальной панели основного интерфейса. Все профили должны быть предварительно сохранены в папке *VPC Link Tool/autoload* , имя отдельно взятого файла должно быть задано согласно шаблону: *[ЛА].led.json* .
- Опция Check for updates. Если данный флажок включен, приложение при каждом старте будет выполнять наличие новой версии, используя портал обновлений Updates

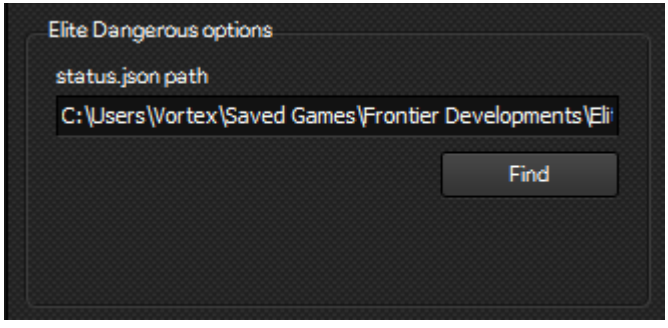
Portal. Если найдена более новая версия – приложение предложит выполнить загрузку установочного файла. Если Вы не желаете получать уведомления о выходе новых версий приложения – данный флажок следует отключить.

Опции Local UDP Address/Port больше не используются и будут удалены в будущем.

Настройки Elite Dangerous

Для правильной интеграции приложения с данным симулятором сначала нужно установить правильный путь к специальном файлу данных *Status.json*. Данный файл содержит различные данные корабля, при этом, сам файл обновляется самим симулятором с высокой скоростью.

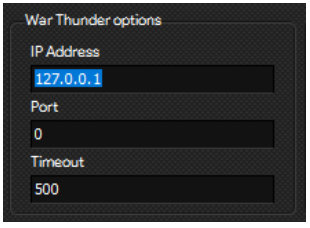
Данный файл можно найти по пути *C:|Users|[YOUR_USER]|Saved Games|Frontier Developments|Elite Dangerous* .



Кнопка **FIND** используется для открытия диалогового окна поиска файла Status.json. После выбора файла в окне – полный путь к файлу будет отображен в поле ввода PATH.

Настройки War Thunder

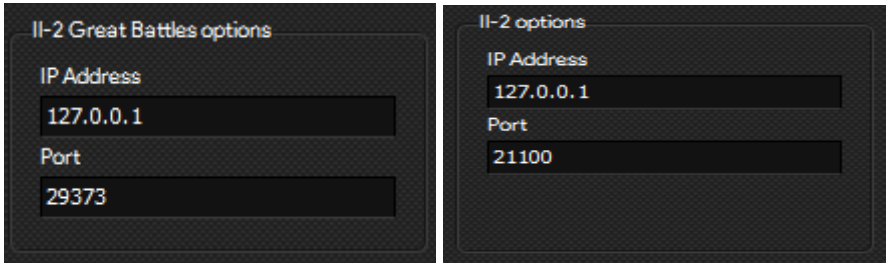
Данная секция используется для ввода данных о локальном веб-сервере War Thunder, предоставляющего данные телеметрии текущего выбранного ЛА в реальном времени.



- **IP Address**. Поле ввода адреса веб-сервера телеметрии War Thunder. Значение данного поля всегда должно быть равно значению адреса localhost сервера (127.0.0.1).
- **Port**. Данное поле не используется и будет удалено в будущем.
- **Timeout**. Данная опция определяет время ожидания приложением данных от сервера телеметрии WT в случае возникновения какой-либо ошибки сервера.

Настройки Ил-2

Данные секции используются для ввода адреса и порта UDP-сервера, который используется приложением для получения данных телеметрии симуляторов семейства Ил-2.

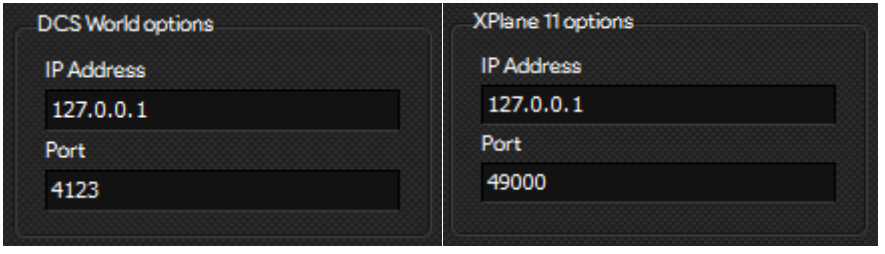


Секция Il-2 options используется для настройки симуляторов Il-2 Sturmovik Cliffs of Dover / Desert Wings.

Более подробно об интеграции приложения с симулятором Ил-2 мы рассмотрим в последующих разделах данного руководства.

Настройки DCS World и X-Plane 11

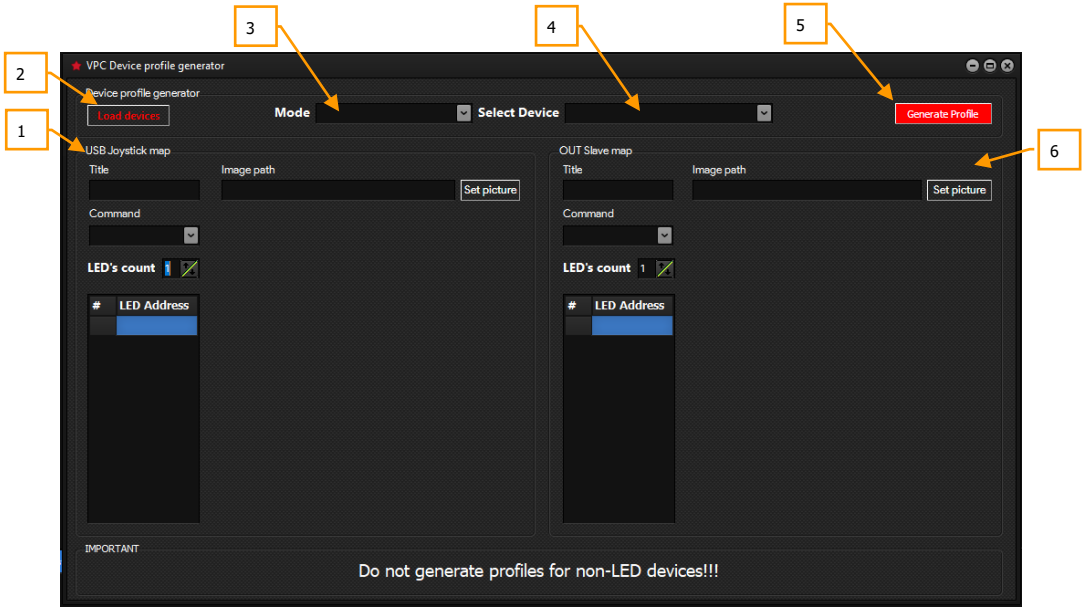
Данные секции используются для ввода адреса/порта UDP-сервера, используемого для получения данных телеметрии ЛА из симуляторов DCS World и X-Plane. В качестве по умолчанию, адрес и порт установлены как 127.0.0.1:4123 для интеграции с DCS World и 127.0.0.1:4123 для интеграции с X-Plane 11. Данные о сервере в приложении должны совпадать с данными в интеграционных скриптах LUA.



Симулятор X-Plane 11 использует, по сути, те же способы передачи данных в приложение, как и DCS World. Более детально мы рассмотрим интеграционные скрипты в следующих разделах данного руководства.

Генератор профилей LED-устройств

Данное окно используется для создания профилей LED-устройств для корректной работы приложения. Каждый сгенерированный профиль определяет количество диодов, расположенных на устройстве (РУС, РУД, панель, и т.п.). Окно состоит из 6 частей:



1. Секция USB Joystick. Данная секция используется для описания основного LED-устройства.
   - I. Поле Title используется для ввода названия устройства, которое выводится в таблице устройств на главном экране приложения.
   - II. Поле Image path используется для ввода пути к вспомогательному изображению устройства, которое выводится при редактировании правила установки цвета диода (в папке assets с приложением в комплекте идут стандартные изображения с адресами диодов).
   - III. Поле Command используется для частичного определения типа устройства и выбора номера команды, которая выполняется приложением для установки цвета диода. Например, если Ваше устройство - РУД, но профиль устройства в приложении содержит некорректный номер команды (к примеру, 101 - РУС), цвета диодов не будут установлены.
   - IV. Поле LEDs Count. Данное поле определяет количество диодов, размещенных на устройстве. Изменяя значение данного поля – изменяется количество строк в таблице, расположенной ниже данного поля ввода. Минимальное количество диодов равно 1.

V. Таблица LED Addresses. Данная таблица используется для ввода адресов диодов устройства, информацию о которых вы можете найти в папке assets приложения.

2. Кнопка LOAD Devices. Данная кнопка используется для загрузки/обновления списка VIRPIL устройств, которые выводятся в выпадающем меню Select Device.
3. Опция Mode. Данная опция используется для определения типа работы устройства. Многие устройства VIRPIL могут работать в 2х режимах: как обычное USB-устройство и дочернее SLAVE-устройство, которое подключено к порту AUX основного устройства, используя кабель типа OUT-AUX. Если основное устройство не содержит дочерних устройств – используйте режим *USB Joystick*. В случае подключения дочернего устройства к основному как SLAVE-устройство, используйте режим *USB Joystick + OUT Slave*.
4. Меню Device. Это меню выбора основного настраиваемого устройства.
5. Кнопка GENERATE profile. После окончания конфигурации профиля устройства нажмите данную кнопку для сохранения/перезаписи профиля устройства. Все профили устройств приложением сохраняются в формате JSON в папку devices. Имя файла профиля выглядит следующим образом: *[PID].json* . Например, *408.json*.

Секция OUT Slave. Данная секция идентична секции USB Joystick и используется для определения опций дочернего устройства SLAVE. Команда для установки цветов диодов любого дочернего устройства - 103, поэтому, окно Command содержит всего лишь одну опцию. Данные этой секции считываются и записываются в профиль устройства только в случае выбора режима основного устройства *USB Joystick + OUT Slave*.

Примечание! Одним из частых вопросов является следующий: "Где я могу найти адреса диодов?". Изображения с адресами диодов находятся в папке assets. Пример изображения РУД:



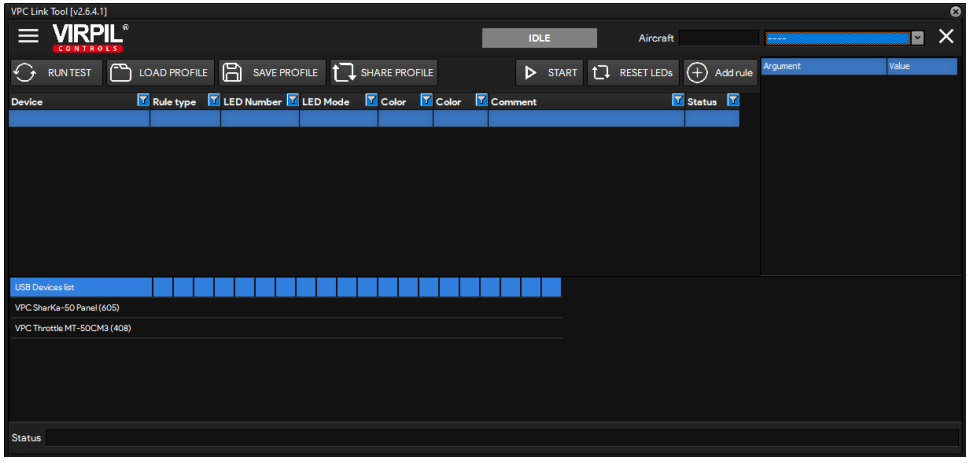
Как показано на картинке выше – РУД CM-2 содержит 6 цветных диодов (выделены красным цветом). Каждый диод имеет свой номер. Данный номер и является адресом диода на плате.

ВНИМАНИЕ!

Не создавайте профили для устройств без LED-диодов (например, педали). Как следствие, приложение будет работать некорректно.

Начало работы

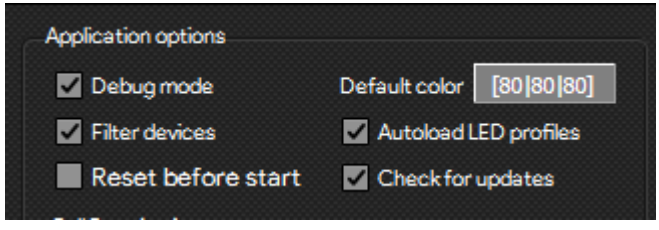
После первого запуска приложения или приобретения нового устройства VIRPIL с LED-диодами нужно создать профиль устройства. Как уже ранее обсуждалось, профиль – это набор дефиниций диодов на устройстве.



Как можно заметить на картинке выше – приложение определило два устройства, для которых не найдены профили диодов.

Для начала – измените цвет диодов по умолчанию, из черного (диод выключен) на серый. Используя боковое меню, нажмите кнопку OPTIONS или нажмите LCtrl + O для того, чтобы открыть меню настроек приложения.

Изменения цвета диодов по умолчанию позволит четко различать части интерфейса. Поскольку приложение выполнено в стиле темной темы – ячейки черного цвета могут сливаться с общим фоном в таблице USB Devices list (Список USB-устройств). Данная опция также устанавливает цвет самих диодов на устройстве по умолчанию до тех пор, пока открыто приложение.

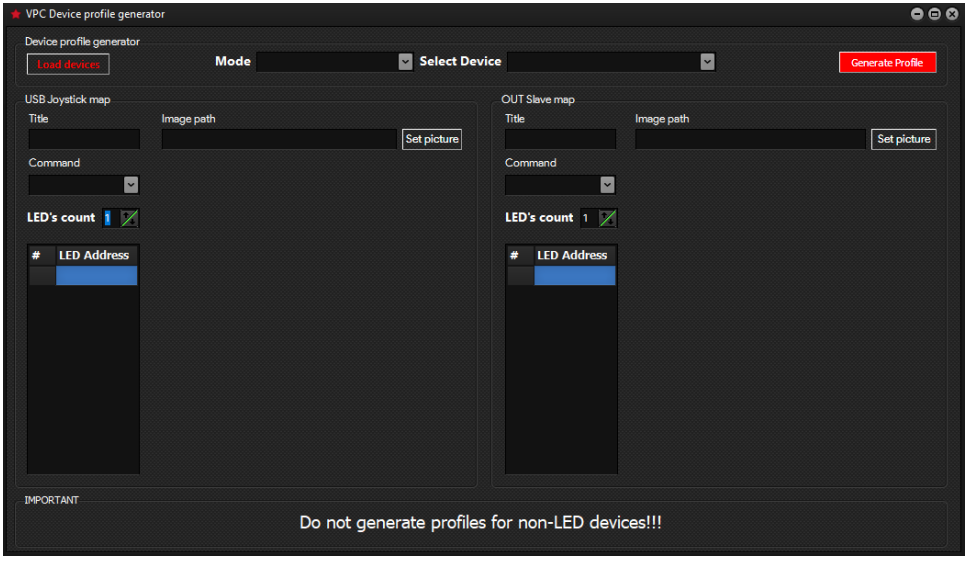


После изменения цвета по умолчанию – перезагрузите приложение.

Следующее, что следует сделать – создать профиль устройства. Откройте генератор профилей нажатием LCtrl + D или нажатием Device Profiles Generator кнопки в левом боковом меню.

На экране появится окно генератора профилей.

Профиль устройства без дочернего SLAVE

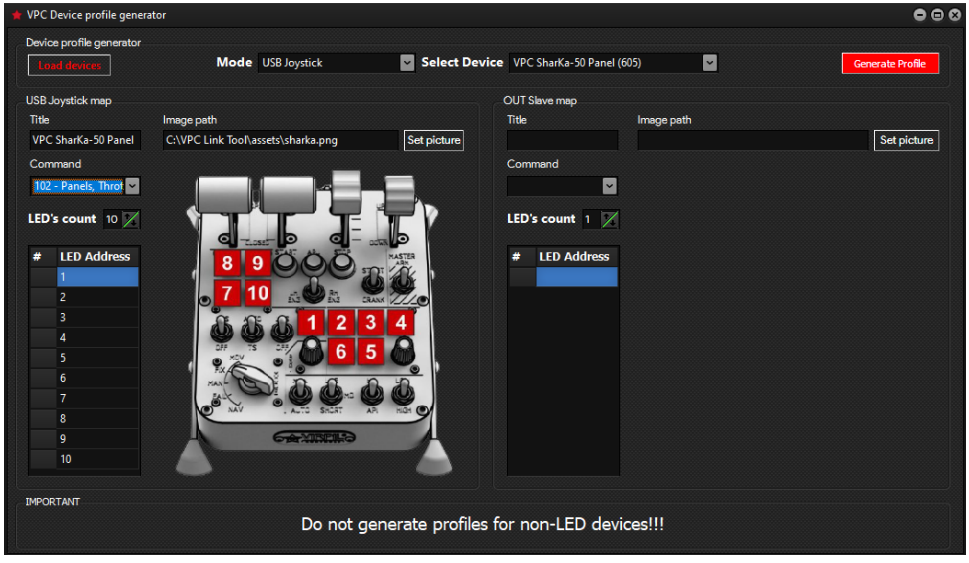


Чтобы создать профиль устройства (в данном случае панель VPC SharKA-50):

1. Выберите режим устройства USB-joystick.
2. Выберите устройство. Используя выпадающий список выберите устройство (VPC SharKA-50). В левой секции окна, ниже поля ввода пути изображения устройства появится изображение красного символа X.
3. Укажите изображение устройства. Используя кнопку Set Picture, в папке assets приложения укажите файл sharka.png. На загруженном приложении представлены номера/адреса диодов на устройстве.
4. Перепроверьте название устройства в поле Title. Значение данного поля заполняется автоматически. Но вы можете изменить название устройства.
5. Укажите значение Command. Если устройство подключено напрямую к ПК без дополнительных дочерних устройств SLAVE – в зависимости от типа устройства – укажите соответствующую команду из списка, которая используется для установки цветов диодов при выполнении запроса в устройство приложением. Команда 101 используется для РУС (Ручки управления самолетом). Команда 102 используется для Панелей/РУДов (рычаг управления двигателем). В нашем случае, панель SharKA-50 требует выбрать команду 102.

6. Укажите количество диодов на устройстве. Используя ранее выбранную картинку в п.3, определите количество диодов. В случае панели SharKA-50 – введите число 10. Приложение автоматически проставит все адреса диодов в таблицу за Вас.

После конфигурирования нового профиля устройства – вы должны увидеть следующее:



Как видно на изображении выше – секция OUT Slave map пустая, поскольку панель SharKA-50 в данном случае подключена напрямую к ПК без дочернего устройства, с помощью кабеля OUT-USB (как USB Joystick).

После выполнения описанных ранее процедур – нажмите кнопку Generate profile. В случае корректного создания профиля – на экране появится соответствующее сообщение. Нажмите ОК. Приложение перезапустится автоматически. После перезагрузки, таблица с устройствами должна содержать 10 ячеек серого цвета напротив панели SharKa, как показано на картинке ниже.



Профиль устройства с дочерним SLAVE

Для демонстрации создания профиля основного и дочернего устройства были выбраны следующие устройства: VPC Panel #2 и VPC Throttle MT-50. Панель #2 подключена к РУД с использованием кабеля OUT-Slave как дочернее устройство для минимизации количества занятых USB-портов на ПК.



Как можно заметить на изображении выше, приложение определило только основное устройство – это РУД MT-50CM3. Панель #2 в списке отсутствует. В то же время, LED-диоды не определены – ячейки темного цвета, сливающийся с фоном приложения, напротив названия РУДа.

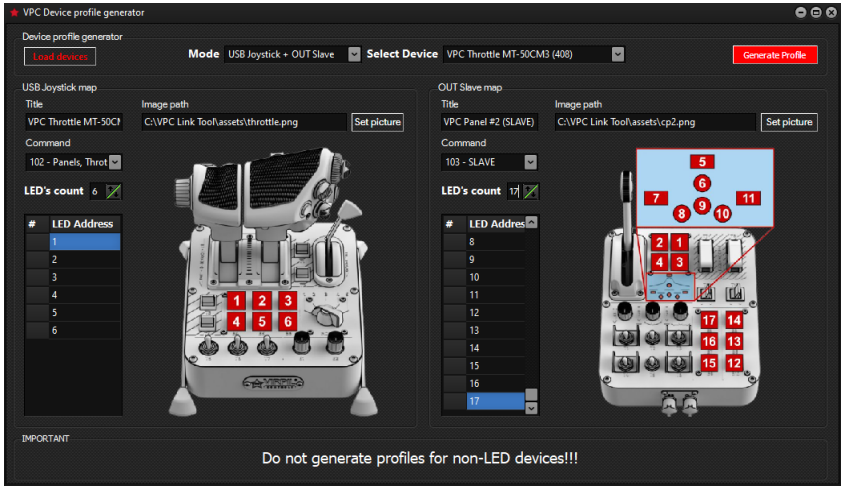
Чтобы приложение правильно определяло устройства и их диоды – нужно создать профиль основного и дочернего устройства.

Откройте генератор профилей нажатием LCtrl + D на клавиатуре.

1. Укажите режим устройства USB Joystick + OUT Slave.
2. Укажите устройство. В данном случае, из выпадающего списка нужно выбрать основное устройство. Это VPC Throttle.
3. Укажите путь к изображению основного устройства.
4. Укажите команду устройства. В данном случае - команда 102 должна быть выбрана.
5. Укажите количество диодов на основном устройстве. Используя изображение из п.3 -- введите число 6 в соответствующее поле. Приложение проставит адреса диодов в таблицу автоматически. После релиза нового устройства, команда поддержки публикует изображения с адресами диодов в случае, если устройство имеет LED-диоды на плате.
6. Перейдите в секцию OUT Slave map. Введите название дочернего SLAVE-устройства. В данном случае, вводится название VPC Panel #2 (SLAVE). Используйте суффикс «(SLAVE)» чтобы “указать” приложению, что устройство подключено как дочернее с использованием кабеля OUT-Slave.
7. Укажите изображение SLAVE-устройства (cp2.png).

8. Укажите команду для дочернего устройства. Все дочерние SLAVE-устройства используют команду 103.
9. Установите количество диодов в поле LEDs Count. Для устройства VPC Panel #2 – количество должно быть равно 17.

После конфигурирования основного и дочернего устройства – на экране вы должны видеть следующее:



Нажмите кнопку Generate Profile. Затем, приложение перезапустится автоматически в случае успешной генерации профиля.

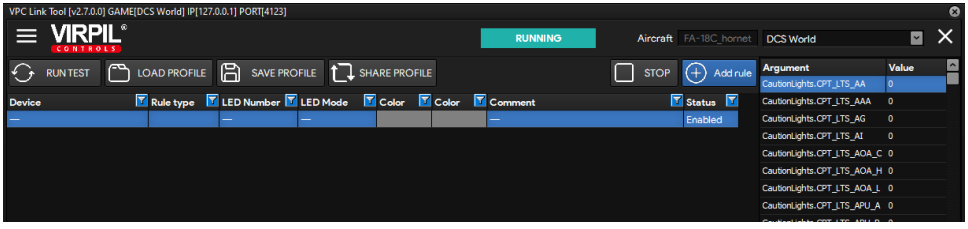
После перезагрузки приложения, таблица-список устройств должна выглядеть следующим образом:



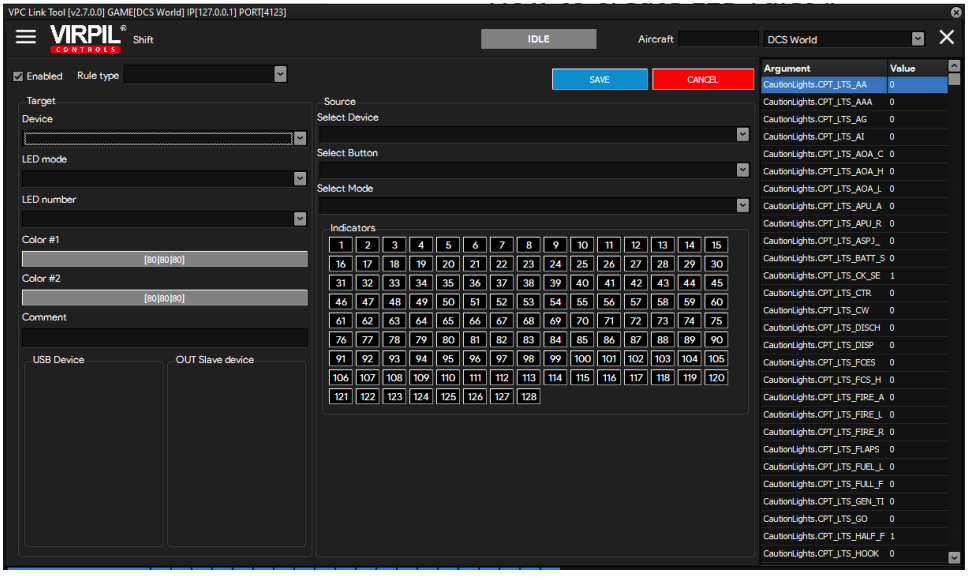
Для проверки правильности созданного профиля нажмите кнопку RESET LEDs, после чего цвета всех диодов на Панели 2 и РУД должны изменится на серый.

Создание LED правил

Для начала – создайте правило-пустышку, нажав кнопку ADD RULE, расположенную в правой верхней части основного окна приложения.



После добавления новой записи в таблицу правил выполните двойное нажатие ЛКМ на новой строке. Появится интерфейс редактирования правила EDIT RULE.

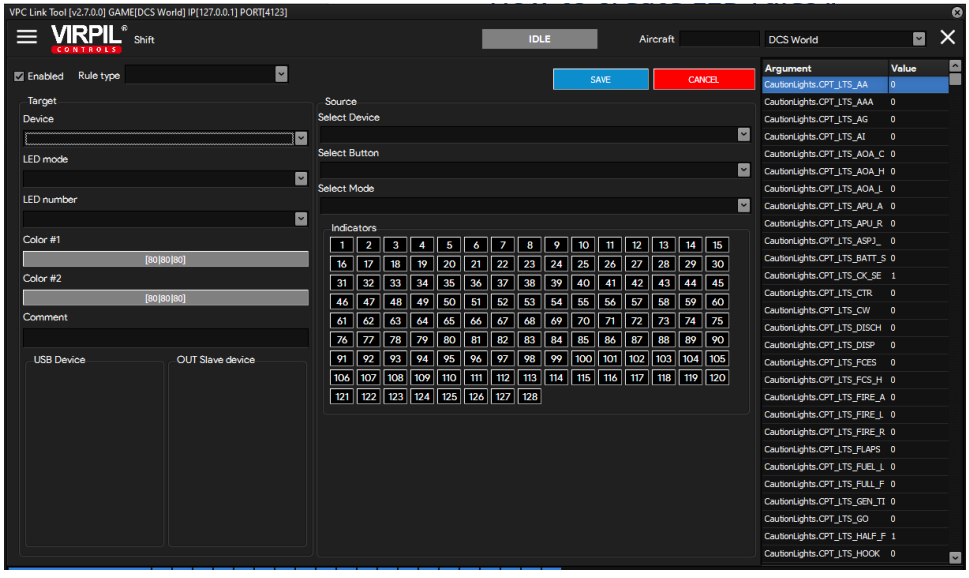


В случае, если приложения работало – произойдет остановка, при этом, таблица аргументов останется заполненной данными телеметрии последнего цикла обмена с симулятором.

Правила аргументов

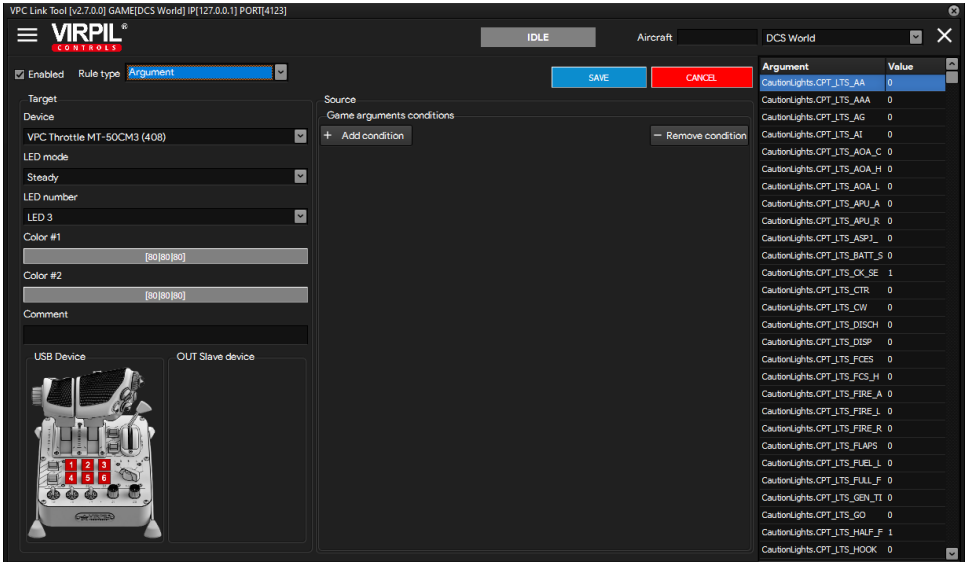
Данный тип правил основан на данных телеметрии симулятора, полученные в ходе циклов обмена с приложением. В данном примере, для демонстрации создания правила постановки цвета диода используется симулятор DCS World (модуль FA-18C).

Симулятор DCS World работает параллельно с приложением. Список доступных данных телеметрии доступен в правой части приложения, в таблице аргументов.



Для того, чтобы создать правило:

1. Выберите тип правила Argument.
2. Укажите целевое устройство Target Device, после чего в секции USB Device/OUT Slave появится изображение устройства. В данном случае, выбрано основное устройство, секция USB Device отображает изображение с адресами диодов на плате.
3. Укажите режим диода - MODE. В зависимости от ваших потребностей, укажите режим Steady («постоянный») или Flashing («мигание»).
4. Укажите номер диода, к которому применится новый цвет и режим в меню LED Number.
5. Установите цвет Color #1. Данный цвет используется в обеих режимах работы диода – Steady/Flashing.
6. Установите цвет Color #2 (если выбран режим Flashing).
7. При желании, оставьте комментарий в поле Comments.



На данном этапе проделана лишь половина требуемой работы. Запрограммировано поведение диода.

Пришло время добавить условия работы диода. Это должно быть сделано в секции Source. По сути, условия представляют собой набор логических операторов «ЕСЛИ», связанных между собой оператором «И». Например:

ЕСЛИ УСЛОВИЕ ИСТИНА

И ЕСЛИ УСЛОВИЕ ИСТИНА

И ЕСЛИ УСЛОВИЕ ИСТИНА

…и так далее.

Для добавления нового условия нажмите кнопку ADD CONDITION. На экране Вы увидите 3 поля ввода информации об условии. НА данном этапе приложение позволяет создать до 13 условий для одного правила.

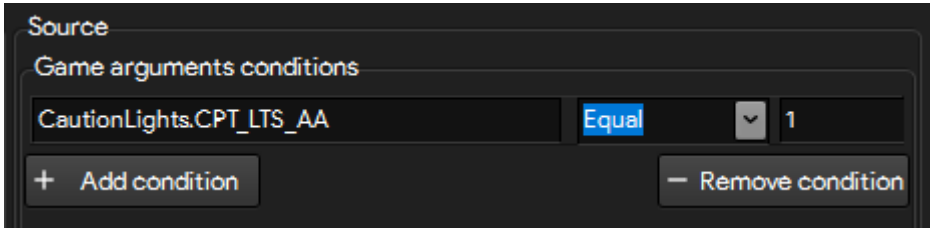


1. Первое пустое поле в левой части строки отвечает за ввод названия аргумента. Это строковое значение названия аргумента, которое выводится в левой колонке таблицы аргументов.
2. Второе поле – меню выбора типа сравнения заданного и текущего значения аргумента. По умолчанию выбран тип сравнения EQUAL (Равенство).
3. Третье поле, расположенное в правом углу строки, отвечает за желаемое значение аргумента, которое будет сравниваться с текущим значением аргумента. В данное поле выполняется ввод любого числового (целочисленного типа или с плавающей точкой float) значения.

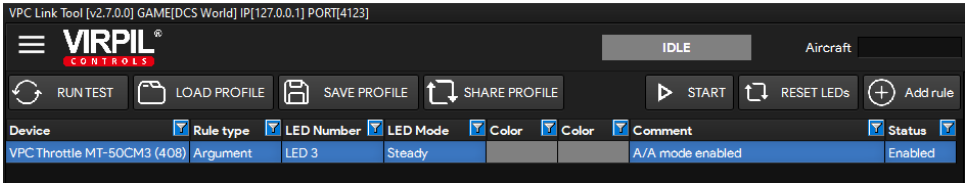
В нашем случае создается правило, которое будет контролировать работу диода LED 3 на устройстве VPC Throttle CM3 в зависимости от состояния режима B-B самолета FA-18C Hornet. Аргумент кабины, который отвечает за состояние режима - *CautionLights.CPT_LTS_AA* .

Теперь нужно данное название вписать в первое поле ввода. У нас уже имеется выбранный тип сравнения EQUAL, далее следует ввести желаемое значение данного аргумента.

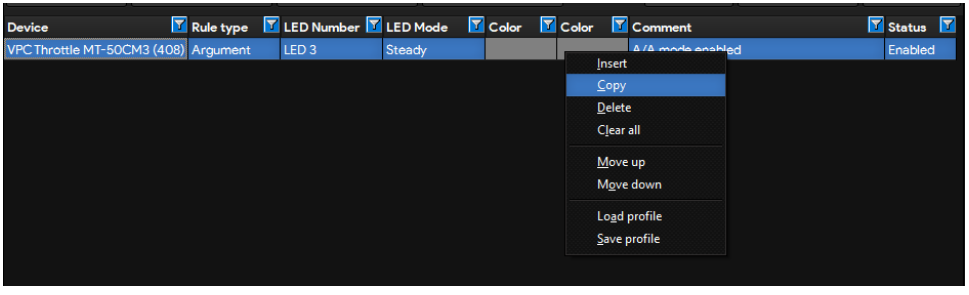
В то время, как DCS World и Link Tool работают параллельно, при включении-выключении данного режима можно заметить, что аргумент *CautionLights.CPT_LTS_AA* принимает значения 0 или 1. Когда режим B-B активирован, в кабине загорается лампа A/A. Такое же поведение ожидается от диода LED 3. Введите значение 1 в третье поле ввода. Результат должен выглядеть следующим образом:

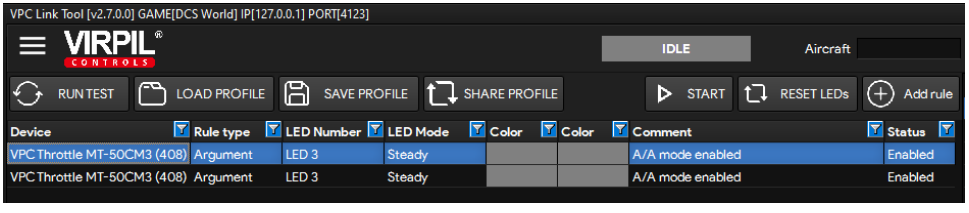


Для того, чтобы сохранить правило, нажмите кнопку SAVE. Вы будете перенаправлены на главный экран приложения.

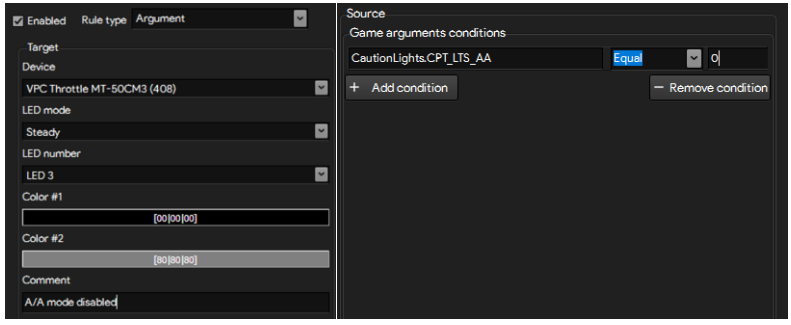


На данном этапе создано правило для включенного состояния лампы. Теперь следует создать правило для отключенного состояния лампы. Для этого вы можете скопировать правило, нажав ПКМ на первом правиле и выбрать опцию COPY (Копировать), после чего – отредактировать новую копию.

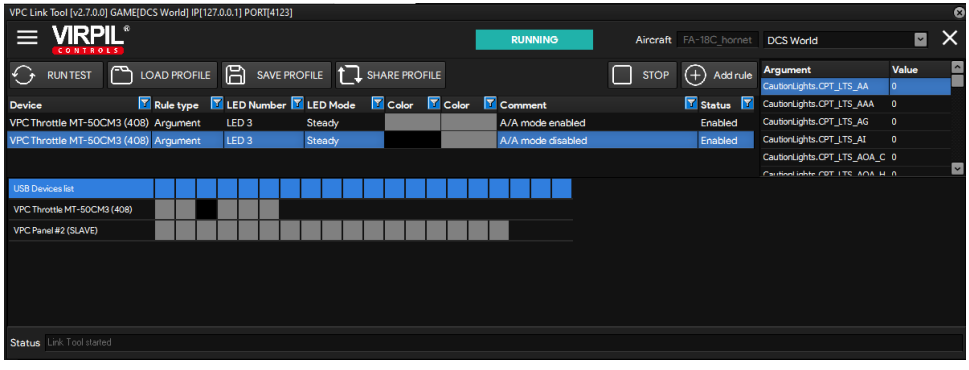




После копирования – отредактируйте второе правило. Двойной клик ЛКМ на втором правиле откроет интерфейс редактирования правила. Измените значение 1 на 0 в третьем поле ввода, а значение цвета Color #1 следует изменить на черный (данный цвет служит для определения состояния диода как Отключен, данный цвет не применяется к диоду). Нажмите SAVE чтобы сохранить второе правило.

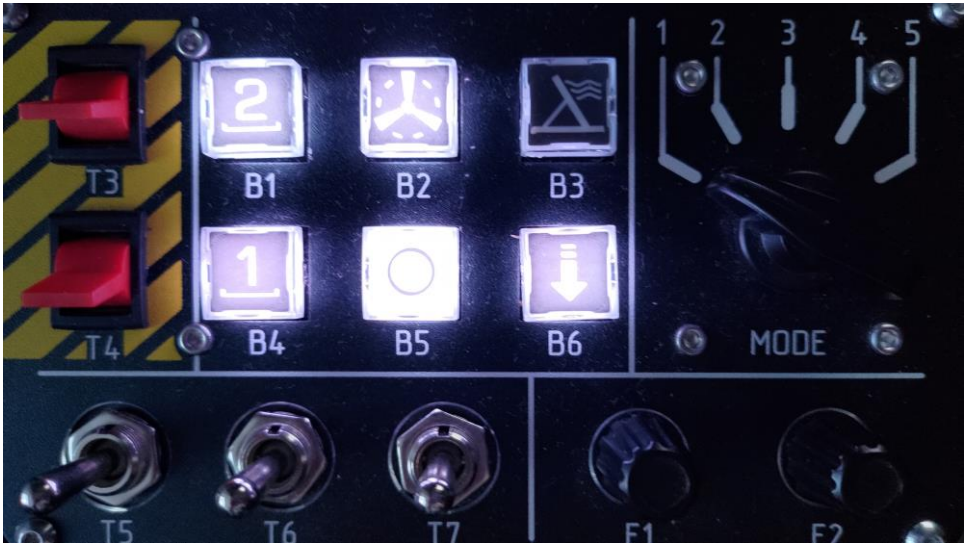


Нажмите кнопку START для запуска приложения в режим работы. Для того, чтобы проверить состояние диода – можно использовать таблицу устройств, которая расположена в нижней части приложения.



Как видно на изображении выше, диод LED 3 на РУДе CM3 выключен, поскольку текущее значение аргумента *CautionLights.CPT_LTS_AA* равно 0.

Как это выглядит на устройстве – вы можете посмотреть на изображении, представленном ниже.


T3
T4
B1
B2
B3
B4
B5
B6
1 2 3 4 5
MODE
T5
T6
T7
F1
F2

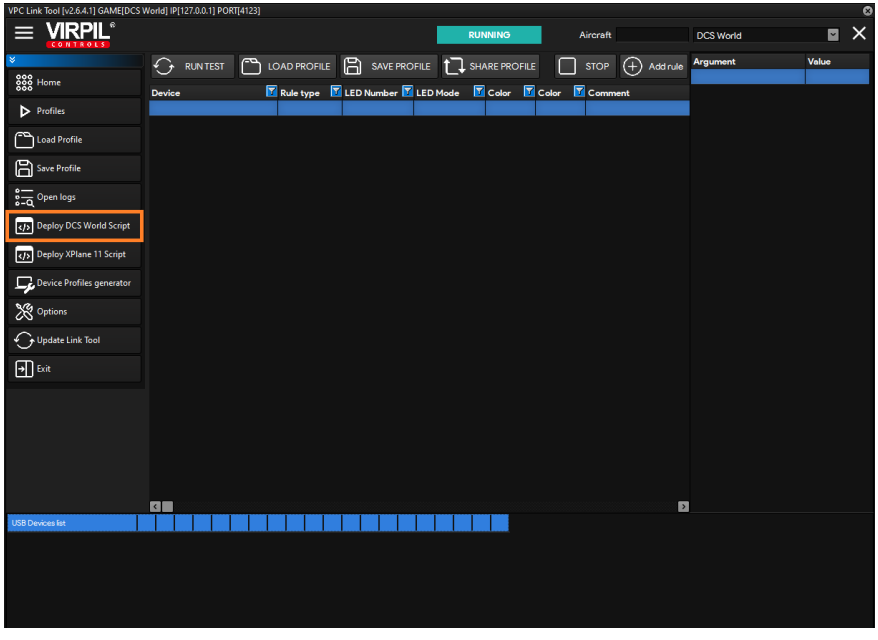
Интеграция с симуляторами

В данной секции руководства Link Tool описана процедура интеграции симуляторов DCS World, X-Plane 11, Microsoft Flight Simulator и Ил-2 с приложением. Некоторые симуляторы для передачи данных в приложение требуют установки дополнительных интеграционных скриптов, которые можно найти в официальном репозитории VIRPIL Controls.

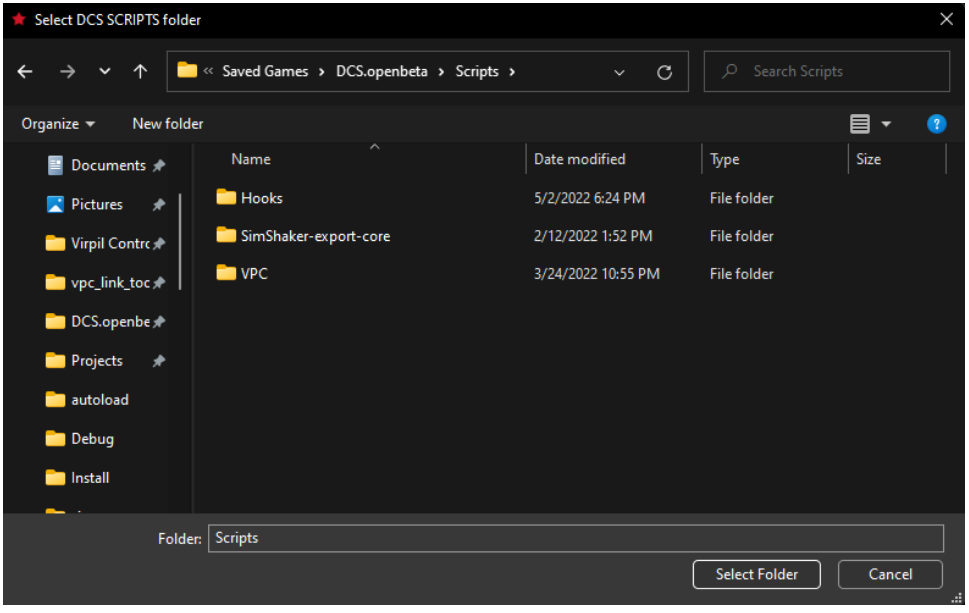
Передача данных из симуляторов DCS World, Ил-2 и X-Plane 11 осуществляется с помощью протокола UDP. Скрипты для DCS/XPlane создают UDP-клиент, задача которого – передача данных из симулятора в приложение. Данные (адрес и порт) для подключения клиента к приложению настраиваются в опциях приложения в специальных секциях, которые были описаны в предыдущих разделах данного руководства.

Интеграция с DCS World

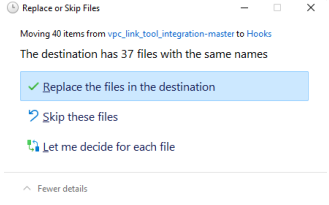
Самый легкий способ интеграции данного симулятора с приложением – это использование функционала Link Tool. В левом меню приложения находится кнопка загрузки скрипта Deploy DCS World Script.



После нажатия данной кнопки приложение потребует выбрать конечную папку установки скрипта *Scripts*, которая, по умолчанию, находится по пути *C:|Users|YOUR_USER|Saved Games*. Если данная папка не существует – нужно создать её.

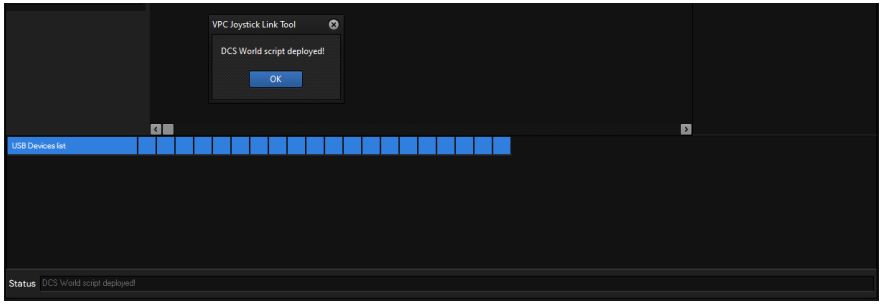


После выбора папки Scripts приложение скачает и установит все нужные файлы в подкаталог Hooks. Если Вы ранее скачивали скрипты – утилита копирования-замены файлов и папок Windows попросит заменить старые файлы на новые.



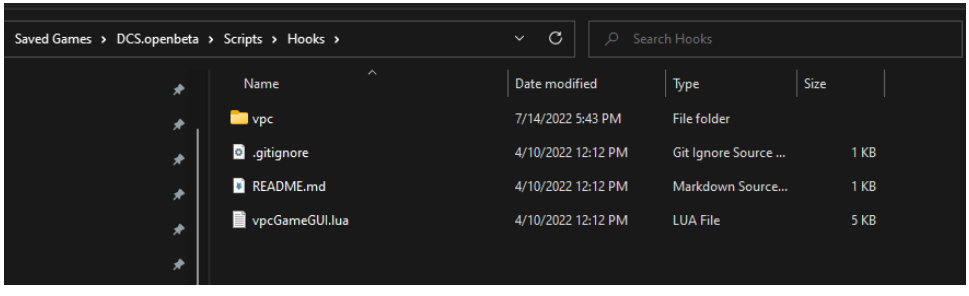
Выберите опцию *Замены*.

После успешной установки дополнительных файлов-скриптов, приложение проинформирует об успешном завершении операции. На экране появится окно с сообщением DCS World Script deployed – Скрипт DCS World установлен. Также, сообщение будет продублировано в нижней строке состояния приложения.



Процесс загрузки скриптов один и тот же для релизной и бета версии DCS World. Единственное отличие между этими версиями симулятора в контексте загрузки скриптов – это конечная папка загрузки и выполнения файлов-скриптов в среде DCS World. Скрипты для бета-версии симулятора устанавливаются в папку *C:|Users|YOUR_USER|Saved Games|DCS.openbeta|Scripts*, для стабильной версии – файлы устанавливаются в папку *C:|Users|YOUR_USER|Saved Games|DCS|Scripts*. В случае изменения папки пользователя DCS World (например, F:\Saved Games\DCS.openbeta) при загрузке скриптов следует выбрать данную локацию вместо стандартной.

После установки скриптов можно рассмотреть структуру файлов. Она должна выглядеть следующим образом:



Немного о файлах. Файл *vpcGameGUI.lua* является точкой входа выполнения кода в среде DCS World. Выполнение данного кода происходит после запуска симулятора.

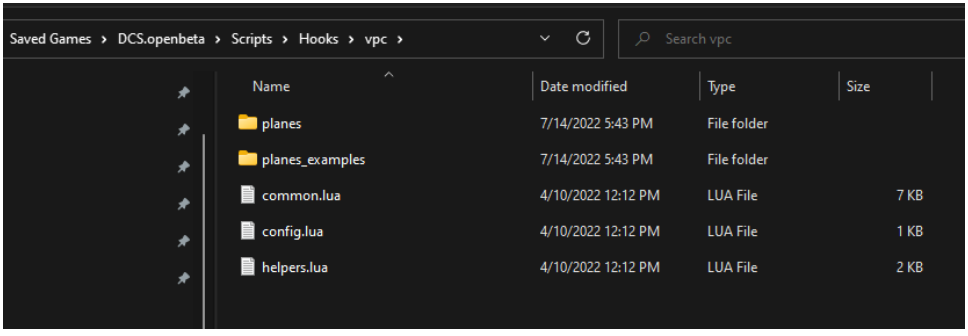
ВАЖНО! Не редактируйте данные файлы, если Вы не знакомы с синтаксисом языка LUA.

По сути, данные, получаемые приложением, разделены на 2 категории (но в конечном счете, объединены в один пакет данных):

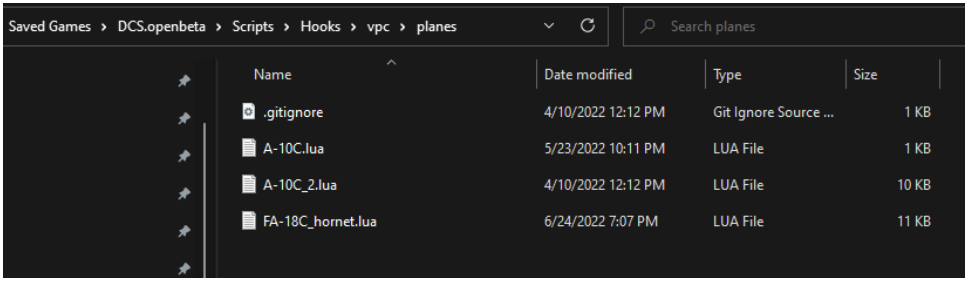
- Общая телеметрия ЛА. Это набор данных телеметрии любого официально выпущенного модуля DCS World, который передается в приложение как в одиночной игре – так и в многопользовательском режиме.

- Телеметрия конкретного ЛА. Это набор данных кабинных аргументов, который может быть определен пользователем путем написания небольших блоков кода. Эти данные, как и общая телеметрия ЛА, передаются в одиночной и сетевой игре.

Важно! Если сервер в сетевой игре запрещает экспорт данных ЛА – общая телеметрия не будет передаваться в приложение.



Первый каталог *planes* содержит файлы экспорта кабинных аргументов для каждого модуля симулятора отдельно. Используя эти файлы можно осуществлять вывод желаемых кабинных аргументов в приложение. Папка *planes_examples* содержит файлы-примеры готовых пользовательских скриптов вывода данных состояния различных органов управления ЛА, транспарантов, переключателей и кнопок кабины. Для того, чтобы использовать эти файлы – скопируйте желаемый файл с названием ЛА в папку planes и запустите симулятор и приложение.



Например, как показано на изображении выше, в папке planes находится 3 файла, отвечающих за передачу кабинных аргументов модулей A-10C, A-10C2 и FA-18C.

Важно! Если скрипт для модуля отсутствует – при первом запуске модуля в симуляторе файл с системным названием модуля будет создан автоматически с помощью других системных функций, заложенных в основной интеграционный скрипт, на основе шаблона-файла

template.lua, который можно найти в папке *Saved Games | DCS.openbeta | Scripts | Hooks | vpc | planes_examples*. Данный файл ни в коем случае не следует изменять-удалять!

template.lua - Notepad

File Edit View

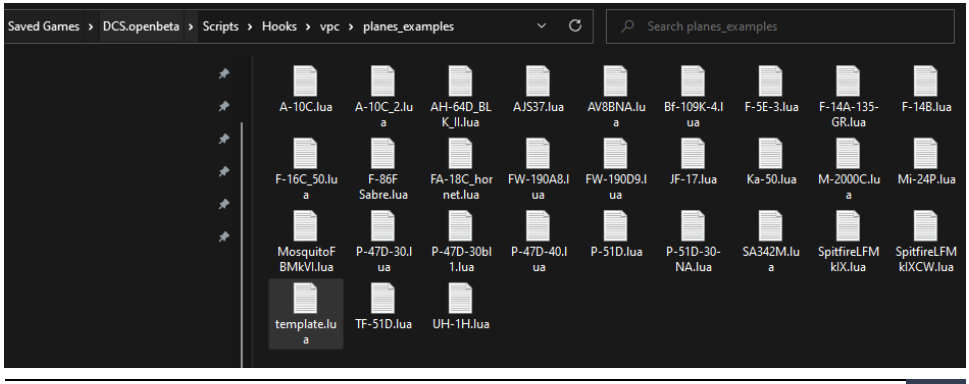
```
return function (vpcPanel)
    local data = {};
    --------------------------- WRITE YOUR CODE HERE -------------------------------------
    -- ARGUMENT_ID - integer value
    -- ARGUMENT_NAME - string value
    --
    -- if (vpcPanel:get_argument_value(ARGUMENT_ID) == 0) then
    --     data["ARGUMENT_NAME"] = "0"
    -- else
    --     data["ARGUMENT_NAME"] = "1"
    -- end
    --
    --
    -- -- OR
    -- data["ARGUMENT_NAME"] = vpcPanel:get_argument_value(ARGUMENT_ID)
    -- data["ARGUMENT_NAME"] = vpcPanel:get_argument_value(ARGUMENT_ID)
    -- data["ARGUMENT_NAME"] = vpcPanel:get_argument_value(ARGUMENT_ID)
    ------------------------------------------------------------------------------------

    return data; -- Don't forget to return data from function in order to display args inside LINK TOOL
end
```

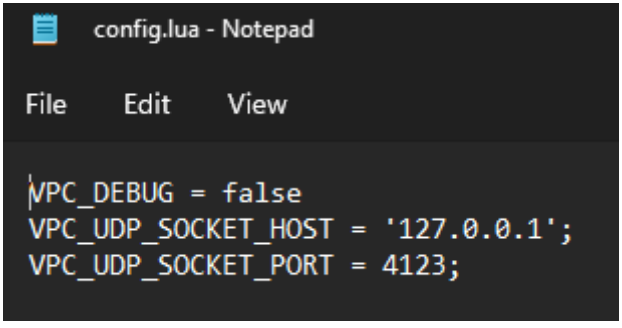
Рассмотрим структуру кода файла модуля экспорта кабинных аргументов.

Задача данного файла – это возвращение анонимной функции, которая принимает в себя параметр *vpcPanel* (объект) и возвращает LUA-таблицу как результат. Все, что требуется от пользователя – это точно знать номер аргумента (его ID - айди) и за что отвечает аргумент. Основываясь на этих данных, Вы можете построить свою собственную логику передачи и отображения данных в приложении Link Tool.

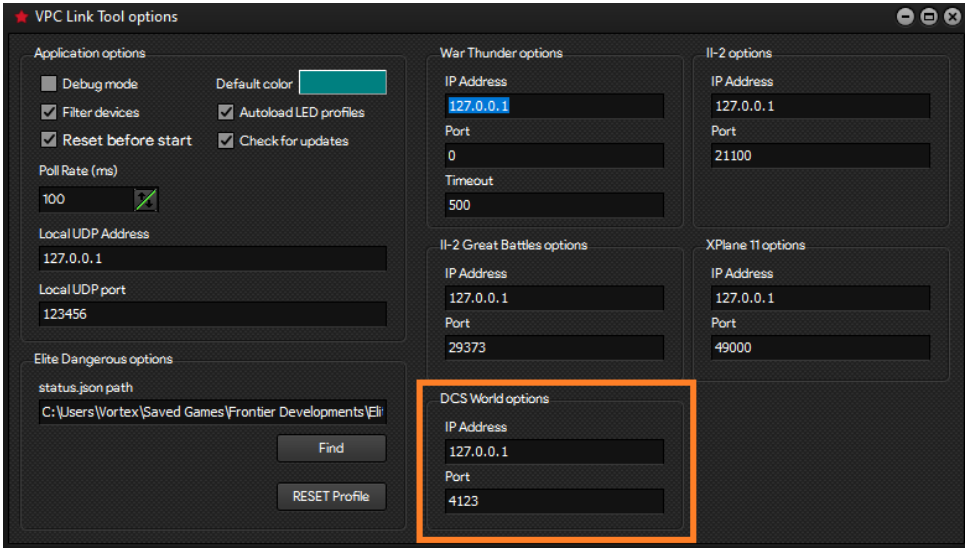
В папке *planes_examples* можно найти множество пользовательских примеров экспорта кабинных аргументов различных модулей DCS World. Экспорт работает как в одиночном, так и в многопользовательском режиме симулятора.



Для того, чтобы понять – получает ли приложение данные от симулятора или нет – стоит взглянуть на таблицу аргументов в приложении. Если таблица пустая – приложение не получает данные телеметрии. В этом случае первым делом следует попробовать изменить настройки UDP-сервера (адрес и порт) приложения и LUA клиента. В папке *C:|Users|YOUR_USER|Saved Games|DCS.openbeta|Scripts|Hooks|vpc* откройте файл *config.lua*.



Следует изменить значение глобальной переменной VPC_UDP_SOCKET_PORT. Порт 4123 нужно заменить на другой, доступный порт. Также, в настройках приложения следует изменить порт в секции DCS World.



Во время редактирования код и/или разработке собственной логики кабинных аргументов полезно использовать опцию VPC_DEBUG. Если значение данной переменной задать как **TRUE**, симулятор будет выполнять перезагрузку пользовательских скриптов при загрузке миссии или

входе на сервер. Если данная опция установлена как False – после изменения кода пользовательских скриптов каждый потребуется полная перезагрузка симулятора.

ВАЖНО! Не используйте VPC_DEBUG=True в повседневной работе, так как данный режим перезагрузки скриптов в среде DCS World используется исключительно для отладки кода. Также, данная опция может вызывать конфликты других сторонних приложений-скриптов для DCS World (например, Simple Radio Standalone или TacView).

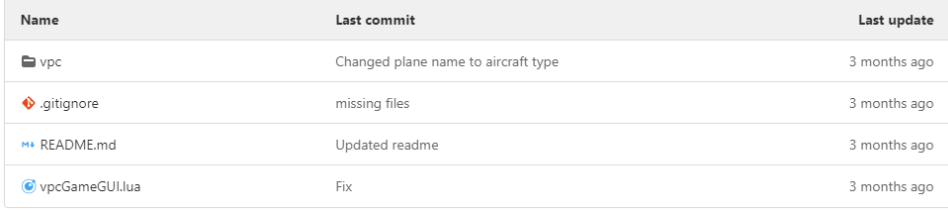
Инструкция, как подключить DCS World к приложениюLink Tool

1. Откройте боковое меню приложения и нажмите кнопку Deploy DCS World Script
2. Укажите путь к папке C\Users\YOUR_USER\Saved Games\DCS\Scripts (Если папка отсутствует – создайте её)
3. Дождитесь загрузки файлов, после чего запустите DCS World.

Существует 2 других способа установки интеграционного скрипта для DCS World.

1. Загрузка кода напрямую с официального репозитория VIRPIL Official Repository используя кнопку DOWNLOAD (будет загружен архив) и распаковка файлов в папку *C|Users|YOUR_USER|Saved Games|DCS|Scripts*.
2. Продвинутые пользователи технологии контроля версий GIT могут «клонировать» код скрипта вручную или с использованием приложений для работы с GIT (Git SCM GUI, Github Desktop, и т.п.) напрямую в папку *C|Users|YOUR_USER|Saved Games|DCS|Scripts*. Данный способ позволит периодически обновлять код скриптов, таким образом – получать последние обновления файлов. Набор команд для клонирования кода следующий:
    - mkdir C:\Users\[YOUR_USER]\Saved Games\DCS.openbeta\Scripts
    - cd C:\Users\[YOUR_USER]\Saved Games\DCS.openbeta\Scripts
    - git clone git@git.dev.virpil.com:dcs-world/vpc_link_tool_integration.git .\

Как уже обсуждалось ранее, структура файлов папки Scripts/Hooks должна выглядеть так:



Name	Last commit	Last update
vpc	Changed plane name to aircraft type	3 months ago
.gitignore	missing files	3 months ago
README.md	Updated readme	3 months ago
vpcGameGUI.lua	Fix	3 months ago

В противном случае, код не будет загружен и выполнен в среде DCS World. Передача телеметрии будет невозможна.

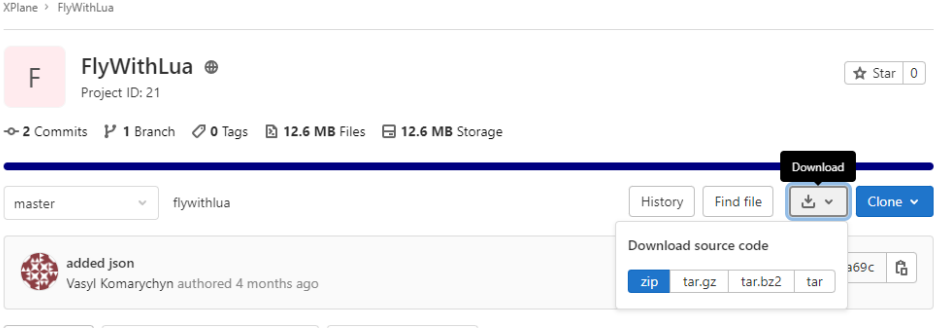
Если у Вас не получается наладить передачу данных в приложение, Мы рекомендуем обратится в техническую поддержку VIRPIL или задать вопрос на официальном форуме VIRPIL.

X-Plane 11 Integration script

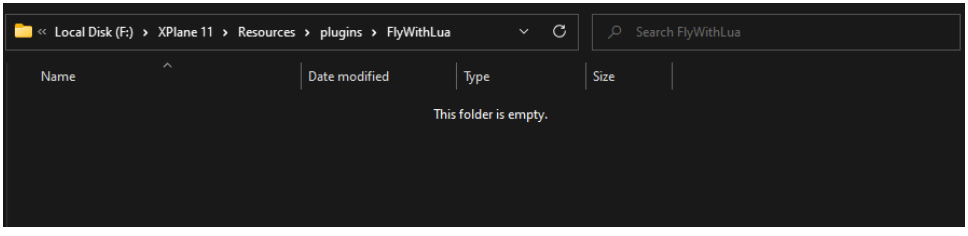
In this section we will cover how to transfer telemetry data from X-Plane 11 to LT. In order to do it using LUA Scripting language, please, download or clone FlyWithLua library from VIRPIL Official Repository.

In case of downloading:

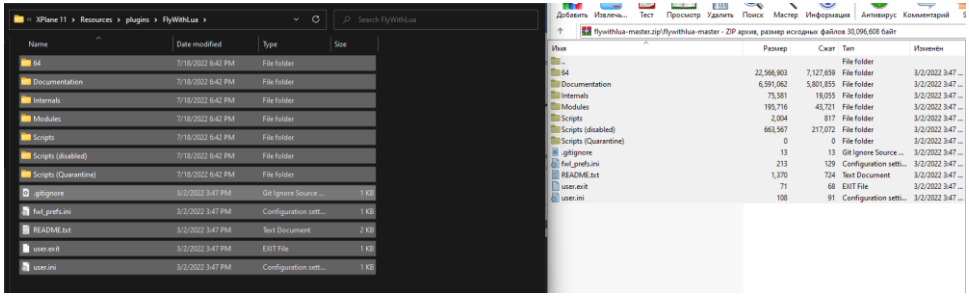
go to https://git.dev.virpil.com/xplane/flywithlua and hit Download button, select ZIP type and wait until archive file will be downloaded.



Go to X-Plane 11 location and create a folder FlyWithLua in XPlane 11\Resources\plugins folder as shown on the picture under.



Unzip all files from downloaded archive (flywithlua-master.zip) into FlyWithLua folder.



In case of cloning project using GIT, open CMD (Command Prompt) and run a set of commands:

```
C:\Users\Vortex>F:

F:\>cd "XPlane 11"

F:\XPlane 11>cd Resources

F:\XPlane 11\Resources>cd plugins

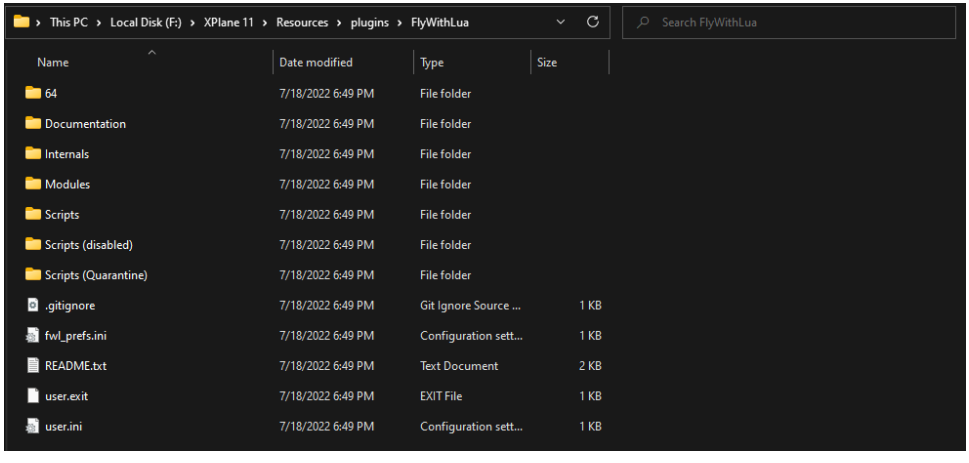
F:\XPlane 11\Resources\plugins>
F:\XPlane 11\Resources\plugins>mkdir FlyWithLua

F:\XPlane 11\Resources\plugins>cd FlyWithLua

F:\XPlane 11\Resources\plugins\FlyWithLua>git clone git@git.dev.virpil.com:xplane/flywithlua.git .\
Cloning into '.'...
remote: Enumerating objects: 188, done.
remote: Total 188 (delta 0), reused 0 (delta 0), pack-reused 188Receiving objects:  89% (168/188), 11.54 MiB | 1.98 MiB/
Receiving objects:  95% (179/188), 11.54 MiB | 1.98 MiB/s
Receiving objects: 100% (188/188), 12.46 MiB | 1.64 MiB/s, done.
Resolving deltas: 100% (13/13), done.

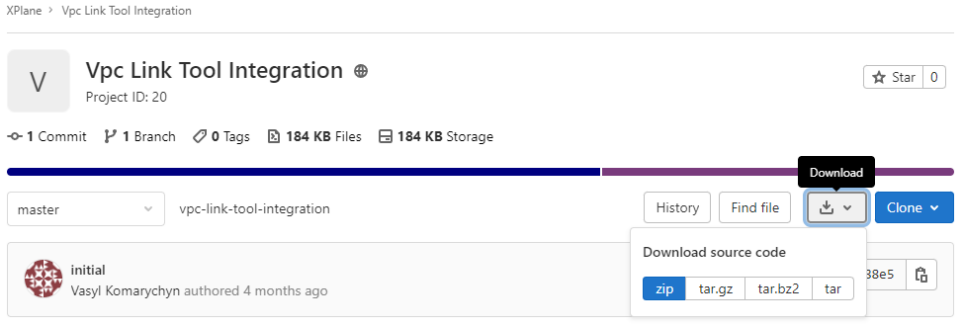
F:\XPlane 11\Resources\plugins\FlyWithLua>
```

Basically, you need move to correct folder, create FlyWithLua folding using MKDIR command and clone project into this folder. At the end, you should see the same folders/files structure:

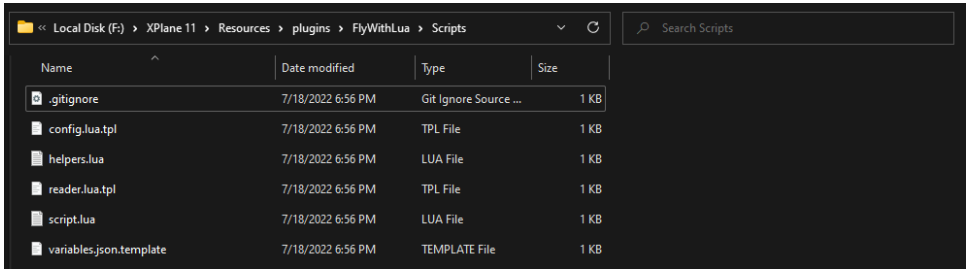


At this point, FlyWithLua library is installed and ready to go. Now, we need to install basic integration script for XPlane 11. This can be done with both download or clone ways from VIRPIL Repository.

Go to https://git.dev.virpil.com/xplane/vpc-link-tool-integration and download ZIP archive of the source code.



Unzip files into XPlane 11\Resources\plugins\FlyWithLua\Scripts so it should looks like on the picture under:



Copy all contents from variables.json.template into variables.json file and your are ready to do some scripting.

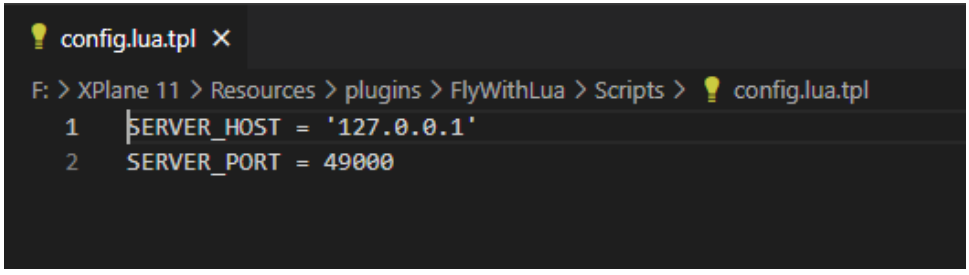
For any further coding we will be using XPlane Datarefs website. It contains a description of all arguments, that are available in XPlane simulator. Go to https://developer.x-plane.com/datarefs/.

As you may see, there is a big list of available categories. But we will discuss a few vars.

Datarefs for X-Plane 1150, compiled Thu May 7 20:05:46 2020

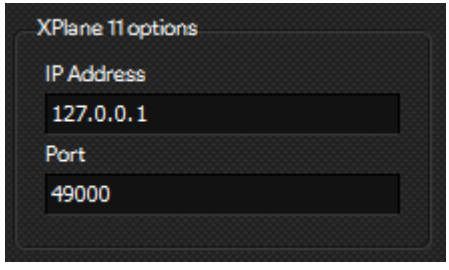
Name	Type	Version	Writable	Units	Description
+ sim/					
+ sim/aircraft/					
+ sim/airfoils/					
+ sim/atc/					
+ sim/cockpit/					
+ sim/flightmodel/					
+ sim/graphics/					
+ sim/joystick/					
+ sim/multiplayer/					
+ sim/network/					
+ sim/physics/					
+ sim/operation/					
+ sim/test/					
+ sim/time/					
+ sim/version/					
+ sim/weapons/					
+ sim/weather/					
+ sim/world/					
+ sim/aircraft2/					
+ sim/cockpit2/					
+ sim/flightmodel2/					
+ sim/graphics/VR/					

At this point lets configure UDP Server data. Open config.lua.tpl file from *XPlane 11|Resources|plugins|FlyWithLua|Scripts* location.

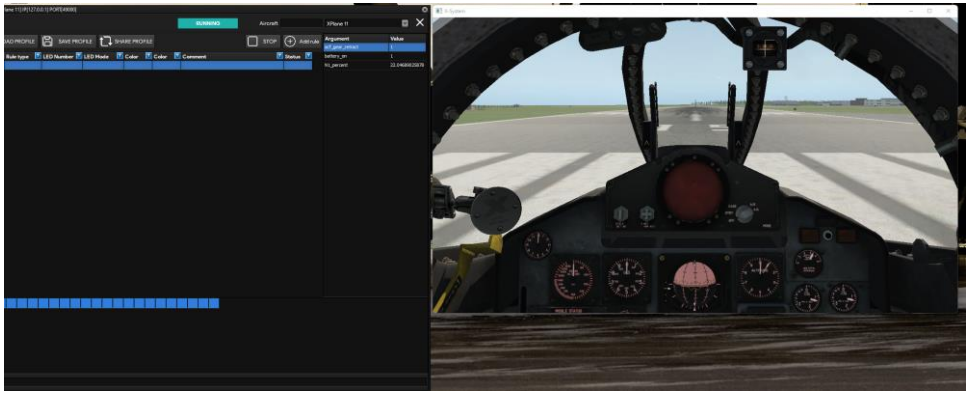


In this file you can find UDP host and port number, to which XPlane will transfer arguments, defined in *variables.json* file. In case, if this file is missing, please, create a copy from *variables.json.template* file.

Configure LT options and set UDP server address and port for XPlane 11 to be the same as set in *config.lua.tpl* file.

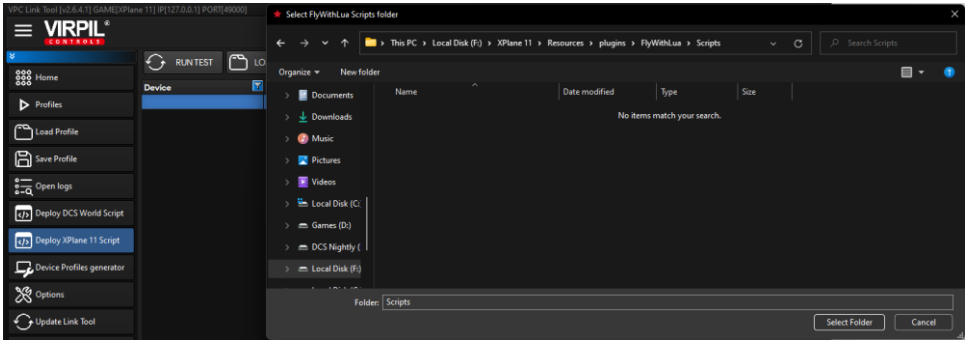


Now, run Link Tool with selected XPlane 11 simulation and run XPlane itself. If you did all steps without any errors, you should see 3 arguments in LT.

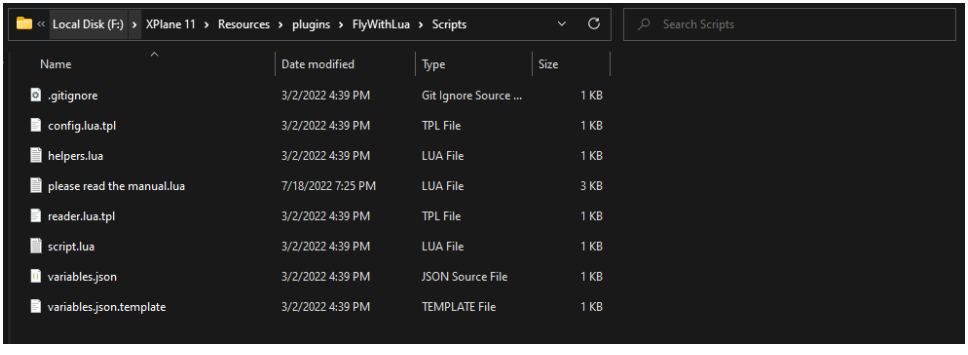


Integration script can be deployed in other way. But you still need FlyWithLua to be installed first.

So, to download integration script, open LT, in the left menu click "Deploy XPlane 11 Script" button, select Scripts folder, which is located in *XPlane 11\Resources\plugins\FlyWithLua* and press Select folder button.



When files will be downloaded, LT will inform you with message : "XPlane 11 script deployed". So, now you can open scripts location and check if files structure is correct.



All files are in place, so you are able to do some magic. Lets see how to add new arguments to LT.

Open file *variables.json* file. This is a place, where all variables are read while XPlane 11 is running.

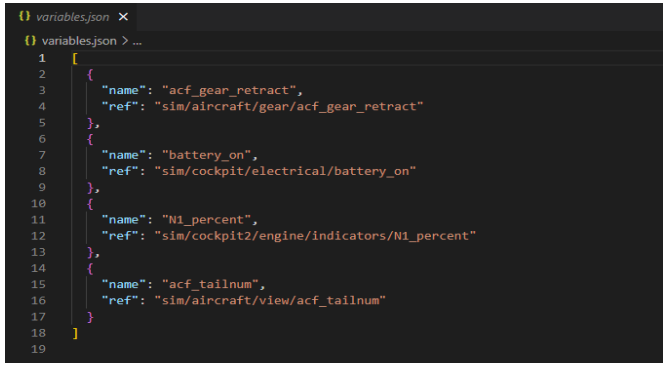


As you can see, on the picture above, content of this file is a JSON Array of objects. Every single object of this array contains 2 keys: name and ref. Key name represents a name of argument from XPlane Dataref table. For example, acf_tailnum.

acf_tailnum	byte[40]	660+	yes	string	Tail number

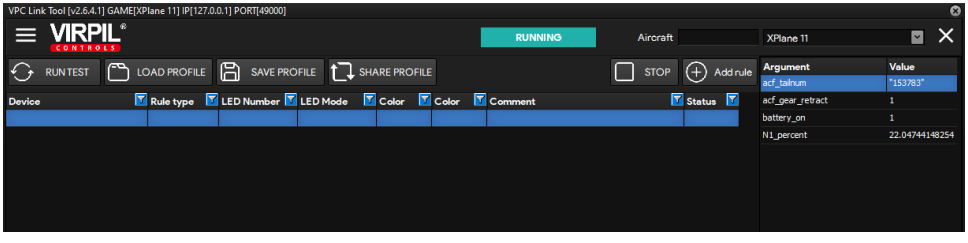
Key ref represents a full path to argument, including category name. In our case, *acf_tailnum* argument is located in *sim/aircraft/view/* category. So, ref value should look like *sim/aircraft/view/acf_tailnum*.

After you found desired argument in Dataref's table, you should define new JSON object in variables.json file. This object in case with acf_tailnum argument should look like :



After making changes in variables file, you can reload all LUA scripts, using FlyWithLua functionality in XPlane 11 in order to see new arguments in LT.





As you can see on the picture above - there is a new argument (acf_tailnum) with value 153783.

You probably, may notice some useless messages on the screen. These messages are generated by manual.lua file, which is located in Scripts folder of FlyWithLua plugin and reload all LUA scripts.

Il-2 Sturmovik simulations family integration

In this section we will discuss LT integration process with a different Il-2 Surmovik versions.

The only exception for LT integration is an Il-2 Sturmovik: Cliffs of Dover simulator. It is ready-to-go for telemetry exchange with LT and does not requires any configuration actions.

Il-2 Sturmovik: Great Battles

In order to connect Great Battles and LT, it is required to change game configuration file called startup.cfg. It can be found in data folder (f.e. C:/Il-2 Great Battles/data/startup.cfg). Open this file using Notepad++ or similar editor.

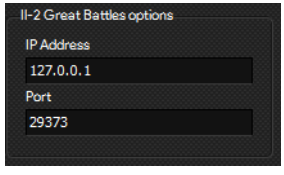
Scroll down and find *[KEY = telemetrydevice]* row. This is a telemetry device section which is used to edit UDP server options, which are used to expose telemetry data from simulator. If this section is missing in the startup.cfg file, just add it manually:

```
[KEY = telemetrydevice]
    addr = "127.0.0.1"
    decimation = 2
    enable = true
    port = 29373
[END]
```

Note. Keep code formatting, so it will be easier to read current configuration file.

Now you can save file and close it. Run simulation. If Il-2 was up and running – restart it in order start/reload telemetry UDP Server, provided by simulation.

After editing startup.cfg file – additional check/configuration is required for LT. Open Link Tool and navigate to options UI using left side menu or using **LCTRL + O** keyboard shortcut. Check or Edit Il-2 Great Battles section in order to match settings, provided for telemetry device in startup.cfg file.



Field `addr` is related to IP Address input field and stands for UDP Server host address. This field always should be as localhost (127.0.0.1).

Field port is related to Port input field. This option can be changed to any available/desired value. But it always should match a value, provided in startup.cfg file. This option stands for UDP Server port, on which Il-2 is broadcasting plane telemetry data.

Possible issues

Sometimes you might face issue, like "Link Tool does not show arguments list". First place you should look at – is a port configuration in both LT Options UI and startup.cfg file.

Another possible issue could be a mess in arguments grid (empty rows or empty values or so). This issue could be related to startup.cfg file. It can contain other configurations for different devices you use in your daily flights. For example, motiondevice configuration section, which is used for SimShaker products like JedPads or similar:

```
[KEY = motiondevice]
   addr = "127.0.0.1"
   decimation = 2
   enable = true
   port = 29373
[END]
```

In this case you may notice, that port 29373 is used for both motion and telemetry device sections. This configuration makes a collision of both data structures (telemetry and motion) when LT reads incoming packets from Il2.

Solution #1: change port number for telemetrydevice in both Il-2 and Link Tool options.

Solution #2: change motiondevice port in startup.cfg and reconfigure SimShaker or other 3rd Party's software you are using along LT in order to use different UDP port.

```
[KEY = motiondevice]
   addr = "127.0.0.1"
   decimation = 2
   enable = true
   port = 29374
[END]
[KEY = telemetrydevice]
   addr = "127.0.0.1"
   decimation = 2
   enable = true
   port = 29373
[END]
```

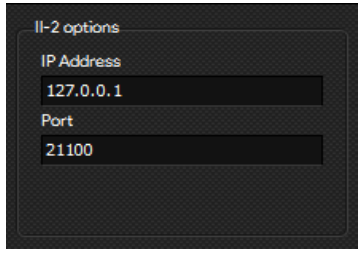
IL-2 Sturmovik 1946 FB

Integration with 1946 FB is similar to the Great Battles.

First, find and open conf.ini file in 1946 FB simulation folder. Scroll down and search for DeviceLink configuration section and edit it. If this section is missing, then add it as follows:

[DeviceLink]
port=21100
IPS=127.0.0.1
host=127.0.0.1

Start or restart simulation. After adding or editing DeviceLink configuration LT will be able to read telemetry data on 21100 port. Check LT options and make sure Il-2 options section is matching DeviceLink configuration.



VIRPIL Devices LED Addresses

